Fast Playback of Helical-scan Recorded MPEG Video

E.D.L.M. Frimout

Fast Playback of Helical-scan Recorded MPEG Video

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College van Dekanen aangewezen,
op dinsdag 28 november 1995 te 13:30 uur

door

Emmanuel David Lucas Michael FRIMOUT

elektrotechnisch ingenieur

geboren te Ukkel, Belgie.

Dit proefschift is goedgekeurd door de promotor:
Prof. dr. ir. J. Biemond,

en de toegevoegd promotor:
Dr. ir. R.L. Lagendijk

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter
Prof.dr.ir. J. Biemond, TU Delft, promotor
Dr.ir. R.L. Lagendijk, TU Delft, toegevoegd promotor
Prof.dr.ir. P.M. Dewilde, TU Delft
Prof.dr. F.C. Schoute, TU Delft
Prof.dr.ir. A.W.M. Smeulders, Universiteit van Amsterdam
Dr. E. Persoon, Philips Research
Dr. L. van Noorden, Commissie van de Europese Gemeenschap

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Frimout, E.D.L.M.

Fast Playback of Helical-scan Recorded MPEG Video /
E.D.L.M. Frimout, -Delft: Technische Universiteit Delft,
Faculteit Elektrotechniek, Vakgroep Informatietheorie. -Ill.
Thesis Technische Universiteit Delft, - With ref. - With summary in Dutch.
ISBN 90-9008561-0
Subject headings: magnetic recording,video coding, fast playback, MPEG

Table of Contents

Preface

The work reported in this thesis forms part of the Digital dAta Recording Terminal (DART) project. The DART project is a collaboration of Philips, Deutsche Thomson Brandt, Grundig and the Delft University of Technology, sponsored by the European Union RACE-II program.

The objective of the DART project is to develop a helical scan recorder that can be used to record the wide variety of audio-visual services that will become available in the future. In particular the helical scan storage technology is attractive to record digital TV broadcasting and IBCN interactive video services which require a high storage bandwidth.

The DART project builds on the results of the RACE-I Digital Video Terminal (DVT) project in which the entire hardware chain of a digital video recorder, including the video codecs, was demonstrated. The bit machine developed in the DVT project forms the basis of the DART recorder.

The activities within the DART project can be divided in two main categories: the recording technology and the adaptation interfacing. In the recording technology a large effort has been made to develop a small recording system, the usage of new materials was investigated and extensive performance studies were carried out. In the adaptation interfacing the requirements for the recording of specific services and applications were studied and some adaptation interfaces were developed. The work of this thesis falls within this interfacing category.

The framework provided by the DART project has focused the work on concrete solutions that form part of the project demonstrators. In this respect the project has had an effect far beyond the initial problem identification and it has assured that realizable solutions were achieved. Consequently this thesis does not present an exhaustive study of possible fast playback approaches. Instead it is constrained to the development and the detailed investigation of solutions that fit within the DART framework.

Summary

The advances in digital video compression and the emergence of hardware codecs conforming to the MPEG video coding standard have been the incentive for the emergence of a variety of new digital video services. In particular the integration of digital video with sophisticated networking facilities will make interactive video applications possible like video database browsing and video on demand. Furthermore it is expected that digital TV broadcasting will have an increasingly important role in the near future. Many of the currently defined digital video services use the MPEG-2 bit stream syntax to represent the compressed video. It can be expected that in the near future MPEG-2 will be the most commonly used representation for digital video.

Within the digital video scenario of the future, the possibility of recording digital video will be crucial. Given the relatively high bit rate of the digital video services, helical scan recording is an appropriate technology. The objective of the DART project, which forms the framework for this thesis, is to develop a helical scan recorder for the wide variety of different digital video services.

From the consumer point of view a new digital recorder should support the well known trick modes like still picture, slow motion and fast playback. Performing a visual fast playback is particularly complicated for compressed digital video. When a helical scan recorder is played back at a speed higher than normal then only portions of the recorded data will be recovered. The physical tape position of these recovered portions is in general not predictable and depends upon the scanner-to-tape phase. These short signal portions from a MPEG compressed bit stream are not necessarily individually decodable and it is not predictable what video sections they represent.

It is therefore necessary to develop a solution to support fast playback for MPEG compressed video recording. An important requirement is that the developed solution must be verified and demonstrated using a hardware demonstrator.

In this thesis we consider the recording of standard resolution video, MPEG encoded at up to 10Mb/s, using a 12.5Mb/s helical scan recorder. For this recording two different approaches to fast playback are defined and evaluated.

In the first approach, which is called rudimentary fast playback, the recovered sections from the normal play bit stream are used to reconstruct a fast playback signal. With this approach most of the complexity of performing fast playback is placed in the playback adaptation interface. The playback adaptation interface has to collect the recovered bit stream sections, select the decodable parts and reconstruct a valid MPEG bit stream that can be decoded by a standard MPEG decoder. In this solution a certain section will be re-used until a more recent version is recovered from tape and the section is updated in the fast playback bit stream.

The visual quality of this type of fast playback is dependent upon the number of individually decodable bit stream sections in the normal play stream and on the size

of the recovered sections from tape. The number of individually decodable sections, i.e. video sections that are encoded without reference to past video sections, determines the average rate at which fast playback picture sections are updated with new information. In the digital TV scenario the recorder will have no influence on the way the bit stream was encoded. The size of the recovered sections from the tape can, however, be influenced and optimized for certain playback speeds. By performing an appropriate mapping of the normal play bit stream on tape longer contiguous bursts can be recovered on fast playback. When longer burst of data are recovered then larger image sections are updated at one time and the visual quality is significantly improved.

The second fast playback approach is that of dedicated stream fast playback. In this approach a dedicated fast playback stream is recorded along with the normal play stream using the spare capacity of the recorder. Instead of using the recovered normal play sections to reconstruct a fast playback stream, the dedicated stream will be used. With this approach the complexity of performing fast playback has been moved from the playback adaptation interface to the recording adaptation interface. Multiple copies of the dedicated fast playback stream need to be written to tape such that, irrespective of the scanner to tape phase, the dedicated stream is guaranteed to be recovered on fast playback. A tape format design methodology for these dedicated streams is developed using a model that predicts the data sections that are recovered as a function of the speedup factor, the scanner to tape phase and a recorder dependent quality factor. A typical example is a speedup of n=3.0 which, for a typical quality factor, requires three copies of the same dedicated stream to be formatted along with the normal play data. The format design methodology can be extended to format multiple streams, each for a different speedup, along with the normal play stream.
The dedicated fast playback stream must be a low bit rate MPEG encoded signal (~1 Mb/s). Many possibilities to transcode this dedicated stream from the normal play stream exist. We propose several codeword extraction methods, where a subset of the codewords of the normal play stream is selected for the fast playback stream. A new codeword extraction method has been developed that approaches the performance of the optimal codeword extraction, while the complexity is only slightly higher than that of the traditional fixed zone codeword extraction method used in literature.

The results of both fast playback approaches have been validated using a hardware verification model of the recorder and the basic adaptation interfaces. In this validation the video encoding and decoding was performed off-line. The adaptation interface formatting and the recording was performed in real-time using hardware interfaces and an experimental bit machine developed by the DART project. For both rudimentary fast playback and dedicated stream fast playback, the results of the validation confirm the simulation studies. In particular, for dedicated stream fast playback the format, which was based on the fast playback model, proved to

perform as expected. The tape format design method is flexible enough to cope with the drift in the playback speed of the experimental recorder by recording one additional copy of the dedicated stream at the cost of retaining smaller bandwidth for this stream.

1. Introduction

1.1 Digital Video Standardization

Recent advances in integrated circuit technology have made the implementation of digital video compression algorithms in real-time hardware an economically viable possibility [Stoj95]. This coming of age of the video compression technology has spurred the development of new digital video services and applications [DVB94, GA94]. These applications range from video conferencing to digital TV, Video on Demand (VoD) and interactive TV. Moreover, personal computers and workstations are becoming important platforms for multimedia interactive applications, like video database browsing. These systems advantageously use an intimate integration of digital video compression techniques with sophisticated network facilities and digital storage media [Feri95].
The main advantage of the migration from analog video to digital video systems is the reduced bandwidth while the same or better image quality can be achieved. For instance, the introduction of digital TV will open the possibility to accommodate many digital video signals within the current channel of a single conventional analog signal in cable or broadcasting applications. This economical advantage of digital video will also drive the development of new services like satellite near-VoD, which transmits one program with different starting times in parallel. At the same time, there is an opportunity to provide a better quality/resolution for existing broadcasting services [GA95].

The emergence of new digital video services have made the standardization of video compression techniques and digital video systems in general, a high priority. Only a standard can reduce the high cost of video compression codecs and can resolve the critical problem of interoperability of equipment from different manufacturers.
Several important video coding methods for specific applications have been standardized by different organizations. For video-phone/video-conferencing applications the CCITT (currently the ITU-T) Study Group XV has standardized a hybrid coding algorithm at bit rates ranging from 64 to 1920 kb/s, known as Recommendation H.261 [H261, Liou91].
For the digital transmission of TV in a video production environment the CMTT/2, a joint committee of the CCITT and the CCIR (currently the ITU-R) on television and telephony, has standardized a coder at 30/45 Mb/s for contribution quality. The decompressed signal of this codec is of high enough quality to be suitable for further processing.
The ISO/IEC JTC1 SC29 WG11, better known as the Moving Pictures Expert Group (MPEG), has standardized a bit stream syntax for the representation of moving pictures and associated audio at a throughput rate up to 1.5Mb/s for Digital Storage Media (DSM). In November 1993 this first phase of the work (MPEG-1)

cumulated in the ISO 11172 standard [MPEG1V]; conceptually it is largely based on the H.261 standard [LeGa91]. In addition to storage media like the video CD, this standard now plays an important role in many multi-media systems.
Recently the MPEG group has finished its second phase (MPEG-2) in which a standard for the coded representation of digital video for a wider range of applications was developed. The MPEG-2 standard will entail an entire family of codecs to be used by numerous applications at a bit rate up to 40Mb/s and beyond. This standard will play an important role in those areas where a higher quality than covered by the MPEG-1 or the H.261 standard is required. Many currently emerging proposals for the broadcasting of digital TV (both standard and high definition) are based upon the MPEG-2 work [Hopk94, GA94, DVB94]. The MPEG-2 standard is thus expected to be at the core of many, if not all, of the digital TV services of the near future.

1.2 DART System Concept

For many of the new digital video services and in particular for digital TV broadcasting, a crucial point for consumer acceptance is recording of this service. The recording of these compressed video services is one of the main subjects of the RACE DART project which forms the framework for this thesis.
The DART recorder can be considered to be a possible digital successor to the analog VCR. It is intended to record the wide variety of digital video services to emerge in the future on a single recorder. Like the analog VCR, the DART recorder is a helical scan tape recorder. Helical scan tape recording yields a large storage capacity and the capability of recording digital services with a high bit rate. The bit machine at the core of the DART incorporates a helical scan mecha-deck, channel electronics and an error correction unit. Different versions of the bit machine are possible; the effective recording bit rate of the respective versions are 50Mb/s, 25Mb/s and 12.5Mb/s.
The concept of the outside world interfacing of the DART is depicted in Figure 1.1. The figure illustrates the convergence of the home entertainment and the multi-media PC environment. The interfaces are defined by the receivers for distributed video services like the cable (CATV), the satellite (SAT) or terrestrial broadcasting. Alternatively the Integrated Broadband Communication Network (IBCN) is considered for the more interactive services such as video conferencing. In this concept the TV the PC and the DART can be in physically very different locations. The recording on the DART recorder should therefore be transparent such that there is no difference between viewing broadcasted digital video directly on reception or recording this video and viewing it at a later instance.

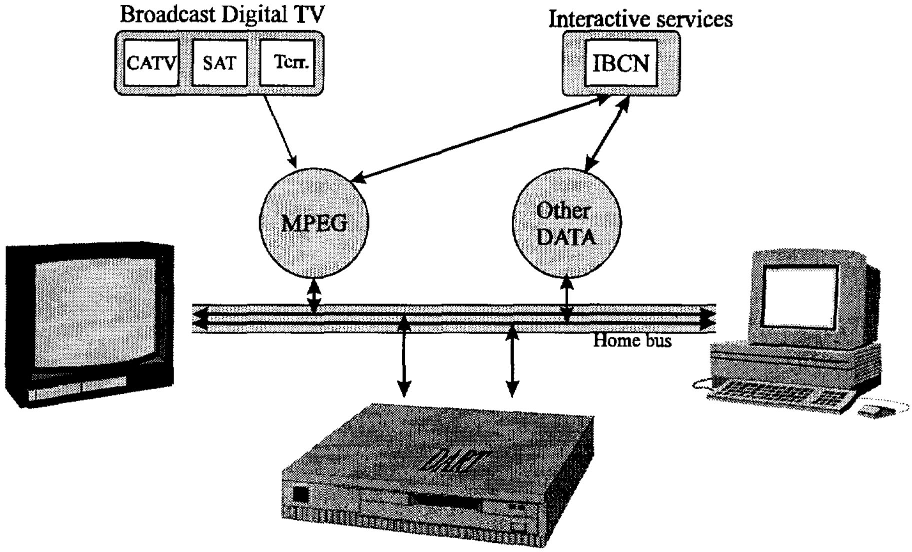


Figure 1.1: DART concept

The objective of the DART is to record in particular those digital TV broadcasting and IBCN interactive video services which require a large bandwidth such as standard resolution TV [CCIR601] with compressed bit rates below 10Mb/s and HDTV with bit rates up to 20 Mb/s [GA94]. As the MPEG-2 standard will be used for many of these services, the DART recorder must be able to record and playback MPEG encoded video in various modes.

1.3 MPEG Recording Problems and Requirements

The recording of MPEG encoded digital video is the subject of study in this thesis. The default approach for the recording of an MPEG encoded signal is to record the bit stream directly to tape without performing any particular formatting. However, a number of problems arise with the direct recording. In this section we will briefly discuss the problems and some of the resulting requirements that rationalize detailed investigations of MPEG recording. The important requirement of fast playback of a recorded MPEG bit stream, which forms the focus of this thesis, is presented in more detail in the next section.

1.3.1 Bit Rates

One of the main characteristics of a helical-scan digital recorder is that the recorder always will operate at a certain maximum bit rate. The digital recorder is required to operate at a bit rate that is sufficiently high to record the intended services. A mechanism must be provided to stuff the video bit rate to the recorder bit rate. At the same time the service providers should comply with this maximum bit rate and refrain from transmitting at higher bit rates. For the recording of MPEG encoded

video some consensus exists on the required bit rate for certain quality levels such that it is expected that an agreement on such a common maximum bit rate can be established.
A complicating factor is the usage of variable bit rates (VBR) for video services. In a VBR stream, peaks much larger than the average may exist. These peaks may be smoothed by the use of a buffer at the recorder input, but still agreement must be reached on the required buffer size and on the maximum bit rate after buffering. Also, the service provider should control the rate such that the peak after buffering will not exceed the maximum allowable bit rate.
The recording of VBR encoded data requires the recorder to stuff this variable bit rate to the constant maximum bit rate of the recorder. On playback, it is imperative that the original variability of the bit rate is restored to assure that the timing of the original bit stream is maintained.

1.3.2 Robustness

In a helical-scan tape recorder a high error rate exists before the error correction unit. The source of these errors are tape damage, head clogging or interference from other signals in the system [Wong94]. The error correction unit is always necessary to reduce the error rate.
An MPEG compressed signal is extremely sensitive to errors. Our investigations have shown that undetected bit errors and burst errors cause severe visual impairment of the decoded picture. The error detection requirements of the recorder are thus extremely high. For a 10Mb/s bit stream a mean time between errors of about 30 hours would be achieved by an undetected error rate below 10^{-12}.
The rate of detected errors usually is much greater than the rate of undetected errors. The visibility of detected errors after MPEG decoding is greatly influenced by the concealment capabilities of the decoder [Jung94, Zhu93, Ghan93]. It is imperative that the decoder has sufficient error recovery and concealment capabilities if a digital tape recorder is to be used in the chain.

1.3.3 Synchronization

In a compressed MPEG signal consecutive frames will use a different amount of bits and consequently, it is not possible to maintain correct timing of the display based on the compressed video signal alone. Special timing provisions by means of time stamps, are included in the bit stream. It is important for the recorder to incorporate this timing mechanism in the recording such that at the decoder a correct synchronized playback is achieved.
The usage of the time stamps will impose timing stability requirements on the recording system; the playback bit rate should be as intended and only limited variation of this bit rate is allowed.

1.3.4 Analog Video versus Compressed Digital Video

In an analog video signal, frames are transmitted one after each other. Each frame comprises 40ms of the video signal and therefore needs 40ms to be transmitted. As

a consequence, when recording an analog signal there is a direct relation between the location of a signal section on tape and the TV frame section it represents on screen.
In a compressed digital video signal the size of a frame in the bit stream domain is in general dependent upon the video data. When a compressed digital video signal is recorded, there will consequently be no relation between the location of a bit stream portion and the picture section it represents.
For the recording of analog video the advantage of a fixed tape section per frame proves to be enormous. For the recording of compressed digital video this lost link will be at the root of most problems met in satisfying the editability and trick modi requirements.

1.3.5 Editability

An desirable feature for consumer camcorder applications is the possibility to edit an encoded bit stream, i.e. replacing N consecutive frames by N new frames directly on the storage media. Several approaches can be taken to satisfy this editability requirement. Many dedicated recording encoders solve this requirement by encoding N frames using no more than a fixed number of bits [McLa93, With93b, Wu91]. An identical approach can be pursued for MPEG encoded bit streams; a specialized bit rate control could assure that a certain target number of bits per group of frames is achieved with a very small error [Frim93, Kees94]. This would make the editing of MPEG compressed bit streams possible.
In the DART concept, however, the video signal is already compressed on reception and the recorder cannot influence the encoder rate control. Therefore very little freedom will remain to support editability. One method would be to use the necessary stuffing in such a way that a group of N frames is stuffed to a fixed tape section size. The complicating factor is that the size (N) of the independently decodable groups of frames may vary in time.

1.3.6 Trick Modi

From the consumer point of view a new digital tape recording system should at least support the well-known trick modes of the existing analog VHS machines, namely:

- still picture
- slow motion
- fast playback (both forward and reverse).

The first two types of trick modi can be accommodated by appropriate design of the recorder electronics. When running the tape at a lower speed or even when halting the tape, all the required information to compose a visual signal is recovered. It will therefore always be possible to reconstruct a still picture or slow motion video respectively.
Performing a visual fast playback search on the recording is however a more complicated matter. When a helical scan recorder is operated at a speed higher than normal, then in general only portions of the recorded data will be recovered. In an analog recorder, a signal portion can be individually interpreted and the frame

section it represents can be deduced based on its position on tape. By reading only certain tape portions it is therefore still possible to reconstruct an entire frame from a number of consecutive frames.
For direct MPEG recording only bit stream portions will be recovered. It is generally impossible to individually decode bit stream portions and determine what frame sections are represented. Basically this means that without special precautions fast playback on directly recorded MPEG is not possible. Special measures will be required to reconstruct a fast playback picture. The desired fast playback solution is for the recorder to generate a fast playback bit stream that any standard decoder can decode. It will be the task of the adaptation interface to ensure that the bit stream is "generic" decodable.

1.4 Fast Playback

Many digital recorders that employ a dedicated (video) codec [With92a, Wu91] mimic the analog behavior for the digitally encoded signal in fast playback. The pictures are sub-divided into blocks and each block is individually encoded using a pre-determined number of bits. By writing the encoded blocks at prescribed locations on the tape it is possible to assure that an entire picture can be build up from the recovered bit stream portions during fast playback.
In case an already MPEG compressed video service is received, a possible approach to still support fast playback is to decode the signal and encode the uncompressed signal with a dedicated recorder codec. The disadvantage of this transcoding approach is that the original form of the compressed data is lost and on playback the stream will have to be transcoded again to be able to transmit a valid MPEG bit stream to the decoder.
Within the DART project the objective is to record the digital video without transcoding the normal play bit stream; on playback the originally transmitted stream should be recovered. It is the intention to do this while maintaining the support of the recording requirements. We will study what measure can be taken to support fast playback of the recorded MPEG signal with particular attention to the tape formatting. In this thesis two different approaches will be evaluated.
The first approach to fast playback of MPEG compressed video is based on the use of shreds of recovered normal play data. The main difference with the dedicated codec methodical approach is that for an MPEG encoded stream it is far less clear what picture sections will be recovered from tape. The resulting trick mode picture will consequently not have a pre-determined quality and every time fast playback is performed a different fast sequence will be reconstructed.
The second approach to a fast playback picture is to use a dedicated fast playback signal added for this purpose to the tape on recording. In literature [Boyc93, Naof93, Yana93b] several examples of this approach can be found where a dedicated fast playback bit stream is put at well chosen locations on tape that can be recovered during fast playback. All these examples employ a bit machine that, during fast playback, is capable of locking on to the specific bit stream portions that contain the dedicated trick bit stream. Our attention will go to a solution that will

work satisfactory with a bit machine that performs no phase locking during fast playback. Guaranteed dedicated stream reading is achieved by writing multiple copies of the dedicated stream at well chosen locations on the tape.

1.5 Validation

An important goal of the DART project is to verify the feasibility of the developed solutions by means of hardware demonstrators. The verification of the recording of MPEG video necessarily involves the recording and real-time playback of an MPEG video signal using an experimental bit machine. Given the various research aspects of the recorder, the core bit machine is not solely determined by the MPEG recording application. In fact, much of this bit machine along with the error correction unit was developed in the predecessor of the DART project, namely the DVT project [With93a].

For the validation of the proposed fast playback approaches a hardware verification model will be used. The verification model, which forms an intermediate step towards the final MPEG recording demonstrator, will involve the real time recording and fast playback of MPEG video bit streams. In this intermediate step the encoding and decoding will be performed off-line. The requirement of performing such a hardware verification imposes particular constraints on the formatting solutions and provides a framework to come to realizable solutions. Nevertheless the developed solutions are generally applicable.

1.6 Outline

The basis of most of the work in this thesis is the MPEG standard. Chapter 2 gives an overview of the MPEG standard such that a clear understanding of the signal to be recorded can be gained. Also, the implications of the standard on a recording system will be identified. In Chapter 3 we perform an analysis of a helical-scan recorder and build a model of the sections recovered from tape during fast playback. This model describes the black box behavior of the recorder. It will be used in the second part of Chapter 3 to develop the tape format design methods and define the formatting for both rudimentary fast playback and dedicated stream fast playback. In Chapter 4 the performance of the rudimentary fast playback is evaluated. Chapter 5 is a study on the possible methods for extracting a dedicated trick mode playback stream from the normal play stream. Particular attention is paid to a solution with a low complexity and a good resulting picture quality. In Chapter 6 the validation of the model based work in this thesis is carried out using the hardware demonstrator. Finally in the discussion of Chapter 7, we will take a broader perspective on the work in this thesis.

2. MPEG framework

2.1 Overview

The ISO-IEC JTC1/SC29/WG11 MPEG group has developed standards for codecs at throughput rates of 1.5Mb/s and up to 40Mb/s, nicknamed MPEG-1 and MPEG-2, respectively [Chia95]. The first one is mainly geared towards Digital Storage Media in a multimedia environment while MPEG-2 has a wider area of applicability. In particular MPEG-2 will play an important role in most of the digital broadcasting SDTV and HDTV services to be implemented in the near future.

This chapter is an introduction to the MPEG-2 standard, which forms the framework for the rest of the thesis. The purpose of this MPEG introduction is to provide a brief overview of the standard and highlight those aspects of the standard that are relevant to the work in this thesis. The important question in MPEG recording is what the structure of the encoded signal is and what encoding choices have been made by the encoder.

In Section 2.2 different parts of the MPEG concept and their relation to actual applications will be discussed. Section 2.3 gives an introduction to the video compression standard and Section 2.4 introduces the system part of the standard. Finally in Section 2.5 the system aspects of MPEG recording will be discussed.

2.2 MPEG Concept

2.2.1 Digital Video Applications

The ISO/IEC MPEG *General Coding of Moving Pictures and Associated Audio* standard [MPEG2S, MPEG2V, MPEG2A] is intended for many different digital video communication and storage applications. Table 2.1 gives an example of some applications that could benefit from the MPEG coding standard.

Many requirements imposed on the compression algorithm and the digital system as a whole, are common to these applications. This justifies the usage of a single standard for all applications. However, some requirements are specific for certain applications. In order for MPEG to be applicable to a large number of applications, all requirements, including the rare ones, must be supported by the standard.

Table 2.1: Typical applications of the MPEG-2 standard
Cable TV distribution
Satellite TV broadcasting
Terrestrial TV broadcasting
Interactive Storage Media (optical/magnetic discs)
Serial Storage Media (digital Video Tape Recorder)
Video on demand
Electronic News Gathering
Network database services (via ATM)
Inter Personal Communications
Remote Video Surveillance

For example, typical flexibility requirements that result from the above applications are [Okub92]:

- Picture Format: The standard supports the three main input formats (4:2:2, 4:4:4 and 4:2:0) both progressive and interlaced, with a flexible number of lines, pixels/line and pixel aspect ratios.
- Bit Rate: The basic target is composite quality within 3-5 Mbit/s and component quality within 8-10Mbit/s for standard resolution [CCIR601] video. For HDTV, bit rates as high as 40Mb/s and more must be possible. Both Constant Bit Rate (CBR) and Variable Bit Rate (VBR) can occur.
- Random Access: Different applications have different requirements for random accessibility. For some applications, like broadcasting, random access should be interpreted as the required time before decoding after tune-in.
- Complexity/Flexibility: The standard is flexible enough to allow for high performance/ high complexity and low performance/low complexity encoders and decoders.

First we will indicate how MPEG attempts to provide a solution to many possible requirements by employing a flexible syntax definition. Next, we will evaluate the implications on the recording requirements for the recorder and to what extent these requirements have been satisfied.

2.2.2 Adaptation layer

The MPEG system philosophy has two layers between the audio-visual signals and the channel.

1. A generic *coding/multiplexing layer*, in which the coding and multiplexing is performed. In this generic layer all those operations are performed that are common to more than one application.

2. An *adaptation layer* for the adaptation to application specific media or channels. This involves specific standards depending upon the medium or channel.

Figure 2.1 shows this system philosophy when applied to MPEG tape recording. All recorder specific operations are incorporated in the adaptation layer. An example is the error detection and error correction. The tape formatting also resides in the adaptation layer. This tape formatting is of critical importance to the subject of MPEG fast playback. The solutions to be proposed to support fast playback will therefore form a (partial) specification of the adaptation interface functionality.

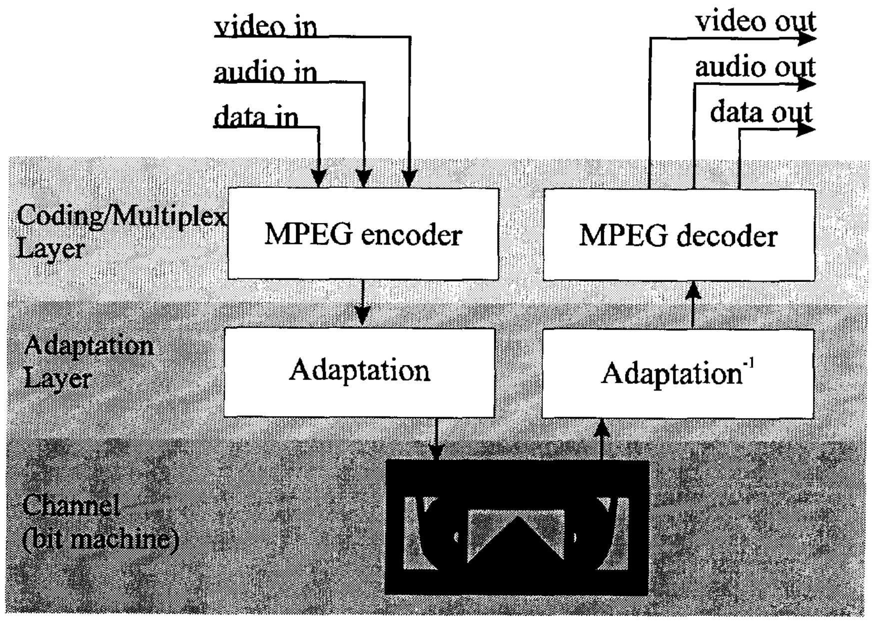


Figure 2.1: MPEG adaptation in a tape recording environment

The adaptation layer is not specified by the MPEG standard and its specification is left to other standardization bodies for specific applications [Look95]. Whatever the behavior of the recording channel may be, the bit stream delivered by the inverse adaptation layer should be guaranteed to have a valid MPEG syntax.

2.2.3 Standard Structure: Video, Audio, System

The MPEG standard contains several separate parts, each of which specifies a separate component. The three main components are the following:

Systems — The MPEG system (standard 13818-1 [MPEG2S]) specifies the multiplex structure for combining audio and video data and a means of representing the timing information needed to replay the synchronized sequences in real-time.

Video The MPEG video (standard 13818-2 [MPEG2V]) specifies the coded representation of video data and the decoding process required to reconstruct the pictures.

Audio The MPEG audio (standard 13818-3 [MPEG2A]) specifies the coded representation of audio data and the decoding process required to reconstruct the audio signal.

The components of the standard only define the bit stream syntax and how to interpret this syntax to reconstruct the synchronized video and audio signals. Obviously this syntax definition implies a global encoder and decoder architecture. Although these architectures do not form part of the standard, in our discussion, we will often use a commonly agreed architecture to clarify the implications of the syntax.

2.2.4 MPEG-2 Toolkit Standard

The goal of the MPEG-2 standard is to be a *generic coding standard* applicable to a great variety of applications and to incorporate the necessary hooks for specific requirements. This goal is achieved by the *toolkit* nature of the standard.
The MPEG-2 toolkit standard has a *maximal common core*. The core is relatively large to assure maximal interworking between the applications. However, care has been taken not to overload the core with features that are required only by some applications. The MPEG-2 core is largely based upon the MPEG-1 coding standard. By surpassing the *constrained parameters* (with respect to picture size, bit rate, buffer size, aspect ratio etc.) of MPEG-1, quite a few of the extra requirements are already satisfied. For some of the application specific requirements some extensions have been made to the syntax.
The simple fact that MPEG-2 has the toolkit approach means that for a specific application a choice of the tools used can be made. In order to assure the interworking among different applications of the same nature, a limited number of subsets of the syntax are stipulated by means of *profiles* and *levels* [Okub93]. A profile is a defined sub-set of the entire bit stream syntax that is defined by the standard. A level is a defined set of constraints imposed on the parameters that still exist within a profile.
A specific profile and level standardizes the syntax of the encoded bit stream, not the compression or multiplex algorithms themselves. Quite a few degrees of freedom remain in the algorithm and it is up to the manufacturer of the encoder to exploit these to optimize the performance of his codec.
For the DART recorder, this diversity in the bit stream syntax, used by different services with different profiles, will be an additional complicating factor. Our main point of attention will be the *Main* profile which is at the core of most currently defined services [GA94, DVB94].

2.3 MPEG Video

2.3.1 Standard Structure

The MPEG-2 video compression standard contains two categories: the non-scalable and the scalable syntax. The non-scalable syntax is a superset of the MPEG-1 syntax, with as main added features the tools for interlaced video and the possibility for added precision.
The scalable syntax forms an optional extension to the toolkit of the non-scalable syntax. Therefore the scalable syntax has the non-scalable syntax as core specification. The usage of the scalable toolkit extensions is highly application specific and has so far not been proposed for any emerging systems.
In the following sections we will discuss the global features of the non-scalable syntax as this forms the core of the toolkit for all applications. The main discussion is based on progressive video, the additions for interlaced video are presented in Section 2.3.8. In Section 2.3.10 we will briefly discuss the features of the scalable syntax.

2.3.2 Basic Coding Scheme

The hybrid coding scheme is at the center of the MPEG video codec. A general structure of a hybrid coding scheme for video sequences is given in Figure 2.2 [Hask72, Nata77, Giro87]. It can be seen that the system is based on temporal DPCM, i.e. each frame is temporally predicted and the prediction error is encoded. Essential in the temporal prediction is the use of motion information.

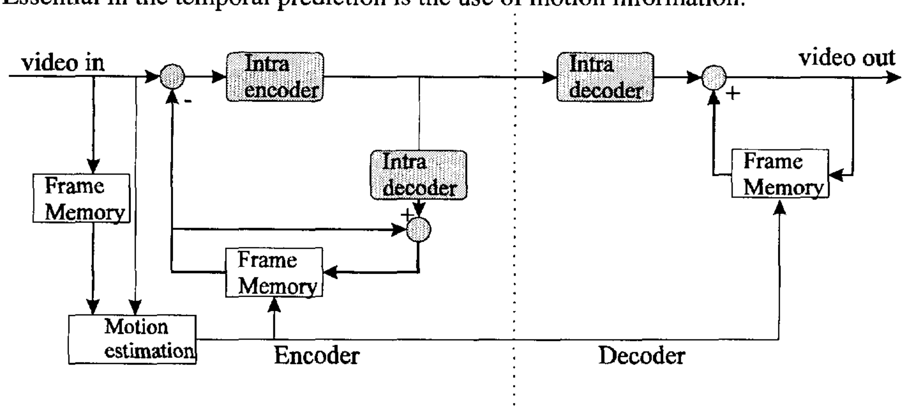


Figure 2.2: Hybrid coding scheme

The motion between a previous and the current frame is estimated [Muss85, Bier88, Dries92, Seza93] and a motion compensated prediction of the current frame is formed. The prediction error, which is called the displaced frame difference (DFD), in general has a much smaller entropy than the original frame. Therefore the intra-frame compression of the DFD can be carried out at a significantly lower bit rate

than if the same intra-frame coder would be applied directly to the frames themselves.
The MPEG compression algorithm relies on block based motion compensation for the reduction of the temporal redundancy and transform domain (Discrete Cosine Transform) based compression for the reduction of spatial redundancy. Spatial reduction techniques are used on the original frame, in case of intra frame coding, or on the DFD frame, in case of motion-compensated coding.
At the receiver the coded prediction error is received and decoded. In the case of motion compensation the motion vectors are transmitted as side information to be used in the prediction of the frame. A frame is reconstructed by adding the decoded prediction error to the prediction.

2.3.3 Layered Syntax

The MPEG video bit stream has a layered syntax [LeGa92]. Each syntactic layer contains one or more subordinate layers. This is illustrated in Figure 2.3. In each layer parameters are coded which define how the subordinate layers should be interpreted. Each layer thus contains a header, with all the relevant parameters, which is followed by the structures of the subordinate layers.
On top of the syntax hierarchy is the *Sequence* layer. At this layer the coding context is defined, i.e. parameters like frame size, frame rate, bit rate and pel aspect ratio are fixed and included in the syntax at this level. A *Sequence* consists of multiple *Group Of Pictures* (GOP) which is a set of pictures which are contiguous in display order. In the bit stream a GOP starts with an intra-coded picture such that it provides a random access entry point. Even if the GOP syntactic header is not included in the syntax, the concept of a group of pictures, starting with an intra coded picture, will prove to be valuable.
A GOP consists of pictures, which are the primary coding units. A *Picture* embodies a luminance frame and two chrominance frames. Depending on the chrominance sampling chosen, they may be subsampled both in the horizontal and the vertical direction.
A *Picture* consists of a number a slices. Each *Slice* consists of a number of macroblocks in raster-scan order. The *Slice* can vary in size. In MPEG-2 the first and last macroblock of a slice will be in the same horizontal row of macroblocks, such that the slice size is limited by the picture width. In the bit stream each *Slice*, and every layer above the slice layer, starts with a synchronization word called a *start_code*. The *Slice* is the lowest layer on which a decoder is able to re-synchronize when synchronization is lost due to an error in the bit stream.
A *Macroblock* contains a square of 16*16 luminance pixels and, depending upon the chrominance sampling structure, at least two 8*8 chrominance blocks. The *Macroblock* is the basic element which is used for motion compensation. Finally, each *Macroblock* consists of six, eight or twelve blocks, depending upon the chrominance sampling. A *Block* is the basis for the intra frame coder.

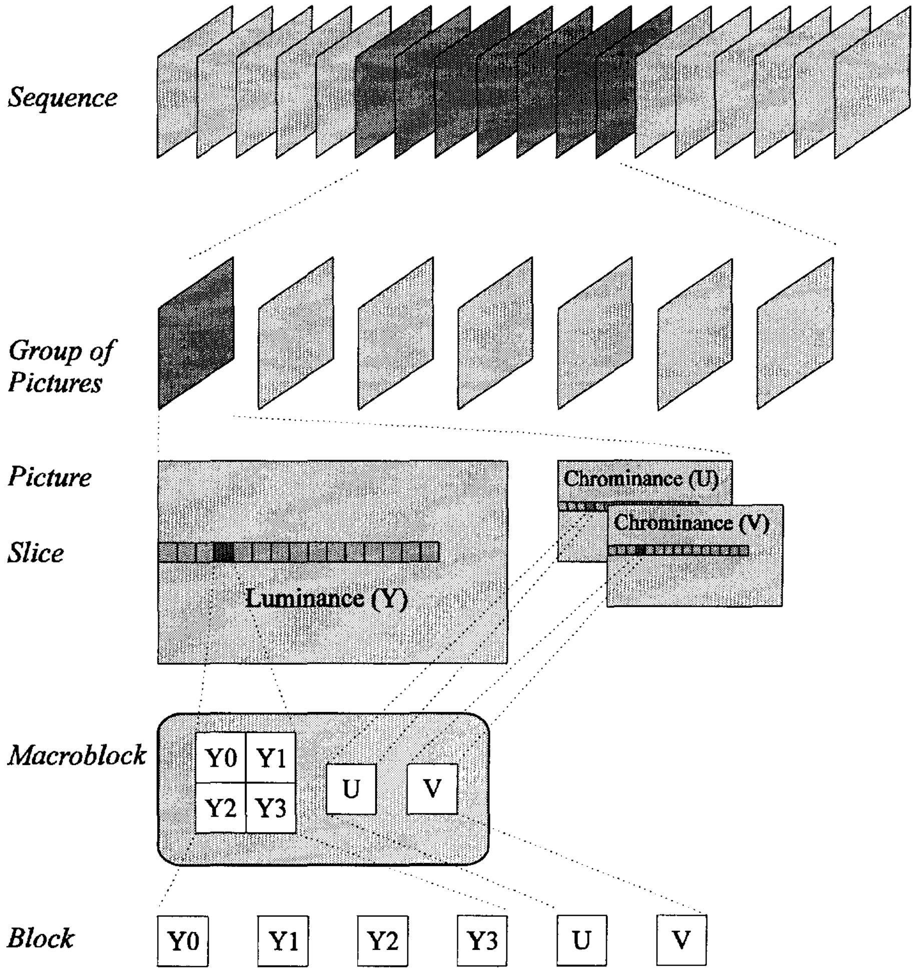


Figure 2.3: Layered syntax

If transmission errors are detected in the bit stream the MPEG standard provides a mechanism for the adaptation layer to signal this to the decoder. An *error_code* similar to the different layer *start codes* can be inserted into the bit stream at the location where the error was detected. When the decoder encounters such an *error_code* it will search for the next synchronizing *start_code*, which will most likely be the next slice, and continue the decoding from there onward.

The layered syntax of MPEG implies that in general it is not possible to decode a lower layer, e.g. a *Macroblock*, if the higher layers are not available. For a recorder performing rudimentary fast playback, using only shreds of recovered information, this is a particularly important observation.

2.3.4 Temporal Redundancy Reduction

Because of the importance of random access for stored video and the significant bit-rate reduction afforded by motion compensated interpolation, three types of pictures are used in MPEG: Intra-pictures (I), Predicted pictures (P) and Bi-directionally interpolated pictures (B). I-pictures, which are encoded without making reference to any other picture, provide access points for random access but yield moderate compression. P-pictures are coded with reference to past pictures (I or P) and will in general be used as reference for future P-pictures and for past and future B-pictures. B-pictures provide the highest amount of compression by allowing prediction or interpolation from past and future references. B-pictures are never used as reference. In all cases where a picture is coded with respect to a temporal reference, motion compensation is used to improve the coding efficiency. The relation between the three picture types in a GOP is illustrated with an example in Figure 2.4.

The organization of the pictures in a GOP is flexible and will depend upon application specific parameters like random accessibility, picture quality and hardware complexity. Note that due to the use of Bi-directionally interpolated pictures, which also reference *future* pictures, the transmission order (bit stream order) is different from the display order. The future pictures to which B-pictures are referenced are transmitted prior to the actual B-pictures. This is illustrated in Figure 2.5 for the example given in Figure 2.4. In the bit stream order a GOP always starts with an Intra coded picture.

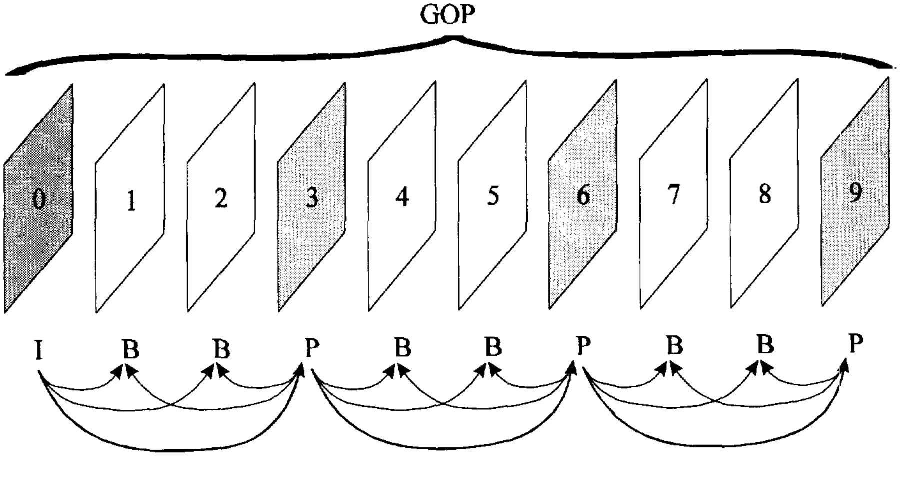


Figure 2.4: A GOP with its three picture types shown in display order and showing the prediction directions

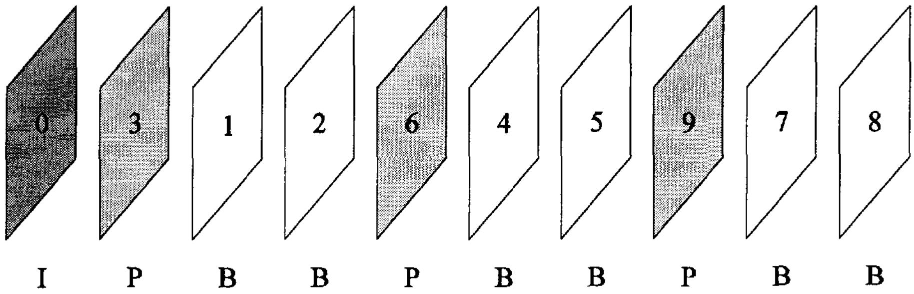


Figure 2.5: GOP of Figure 2.4 with its three picture types shown in transmission order

Although the encoding of a macroblock depends upon the picture type, it is not strictly determined by this. Macroblocks from an I picture form the simplest case; they are all intra coded. For P and B pictures the result of the motion estimation process is evaluated to see if it is appropriate to use motion compensation. For a P picture the macroblocks will in general be coded predictively, i.e. coding the Displaced Macroblock Difference (DMD). The encoder can, however, also decide to intra-code a macroblock if this is considered favorable. For a B picture, each macroblock can be one of the following types: Intra, Forward predicted, Backward predicted and Bi-directionally interpolated.
For every macroblock, a macroblock type parameter in the bit stream will indicate how it is coded. The motion information consists of one vector for forward predicted macroblocks or backward predicted macroblocks, and of two vectors for bi-directionally interpolated macroblocks. The motion associated with each macroblock is coded differentially with respect to the motion information present in the previous adjacent macroblock. The range of the differential motion vector is selected in the *Picture* layer of the syntax.

2.3.5 Spatial Redundancy Reduction

Both the prediction error signals and the original picture contain spatial redundancy which should be reduced for efficient compression. In the MPEG hybrid coding scheme the Discrete Cosine Transform (DCT) [Ahme74] and quantization in the transform domain are used for the reduction of the spatial redundancy. Subsequently a variable length encoding of the quantized DCT coefficients is performed.

- ***DCT transform***

The DCT transform is performed on the 8*8 image blocks. The advantage of using the DCT transform is that it de-correlates the coefficients. Furthermore, for a typical image block, it allowes the energy to be concentrated in the lower frequencies. As the eye is less sensitive to quantization of high frequencies, the high frequency components can be quantized relatively coarse. As such, performing the

DCT allows a large amount of compression to be performed in the subsequent quantization step.

- ***Quantization***

It is the combination of quantization of DCT coefficients and run-length coding of zeros which is responsible for most of the compression. It is also through quantization that the encoder can match its output to a given bit rate; quantization is the key mechanism to achieve an appropriate rate and distortion combination.
MPEG uses a quantization profile for the block coefficients which is defined by a quantization matrix. The ideal *quantization matrix* depends on many external parameters such as the characteristics of the intended display, the viewing distance and the amount of noise in the source. It is therefore possible to design a particular quantization matrix for an application or even for an individual sequence. A customized matrix can be added to the bit stream in the *Sequence* layer or in the *Picture* (*extension*) context. For applications where this is not necessary, a generic quantization matrix has been designed which will be used as the default matrix.

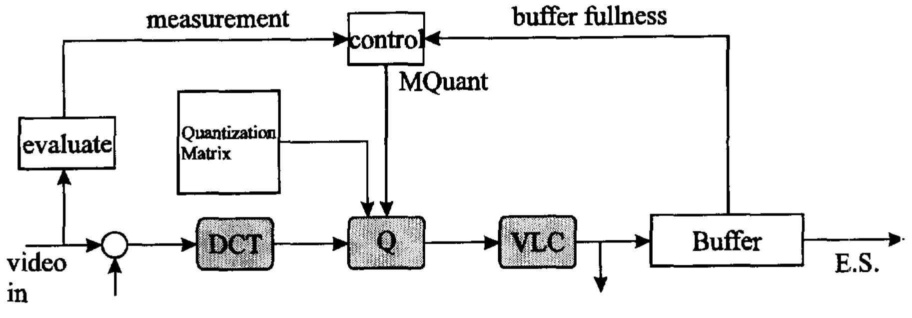


Figure 2.6: Intra coder and buffer with rate control

Figure 2.6 shows the intra coder section of Figure 2.2 with a basic rate control mechanism. The output from the buffer is the compressed elementary stream (ES). The quantizer parameter (MQuant) is used both for rate control and for subjectively adaptive quantization. As a rate control tool the quantizer can be varied by means of a simple output buffer fullness feedback mechanism, to ensure that the video sequence is compressed to the target bit-rate; when the buffer is full coarse quantization is required, when it is empty a fine quantization is performed. As a subjective quantization tool, the quantizer can be either refined or coarsened on the basis of the visibility of impairments for a given block. To this effect an evaluation resulting in a classification of each macroblock into flat areas, edges and different degrees of textures [Puri91] or a classification according to the motion content [Gonz91], can be useful.
The parameter MQuant is normally fixed and transmitted at the *Slice* layer. However it is possible for the encoder to include an alternate MQuant step size for a specific macroblock in the *Macroblock* layer. Which one of these two options is used will be flagged in the *Macroblock* header.

- ***Zigzag re-ordering, run-amplitude pair generation***

Quantized coefficients are ordered along a zigzag path (Figure 2.7) and runs of contiguous zeros are identified. All the non-zero coefficients are now converted into (run, level) tuples, where the run is the number of zeros preceding the coefficient, and the level is the value of the coefficient. The combined (run, level) tuple is transmitted to the decoder.

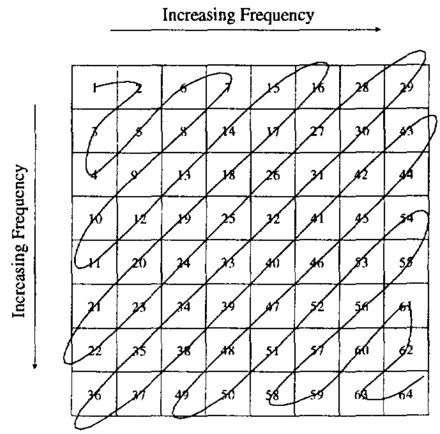


Figure 2.7: Zigzag scan

For a particular block, all occurring (run, level) tuples are entropy coded and transmitted sequentially. After the last (run, level) tuple, a designated End Of Block (EOB) word will be transmitted. When a certain block, from a predicted macroblock, contains only zero coefficients after quantization, it does not have to be coded at all. A Coded Block Pattern (CBP) word will be included in the *Macroblock* layer to indicate to the decoder which blocks have been coded, and which blocks have not been coded.
When none of the blocks of a macroblock are coded, then the macroblock gets the type designator *not coded*; only the motion vectors will be transmitted.

2.3.6 Entropy Coding

For almost all transmitted symbols like differential vectors, (run, level) pairs, macroblock coding types, coded block patterns etc., entropy coding is applied, generating Variable Length Code (VLC) words. For each of the symbol types a pre-defined VLC table exists. The tables have been designed on the basis of a large variety of sequences.
Of the (run, level) pairs only those events with a relatively high probability are coded using VLC words. To avoid extremely long codewords and to reduce the cost of implementation the less likely events are coded with an escape symbol followed by fixed length codes.

2.3.7 Encoder/Decoder Architecture

The described syntax and tools of the MPEG standard can be summarized in a decoder architecture like the one shown in Figure 2.8 [LeGa92]. The bit stream is de-multiplexed into overhead information as the quantization step size, macroblock coding type, motion vectors and (run, level) tuples. After reconstructing a block, the DCT coefficients are de-quantized and input into the Inverse Discrete Cosine Transform (IDCT). The reconstructed block is added to the result of the prediction. Because of the particular nature of Bi-directional prediction, two references are used to form the predictor.

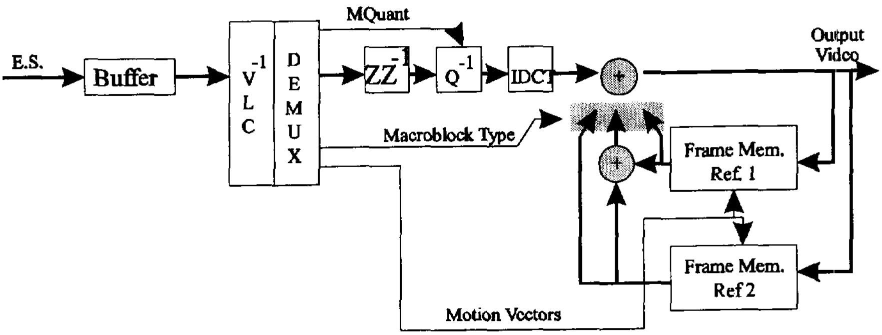


Figure 2.8: Schematic block diagram of the decoding process

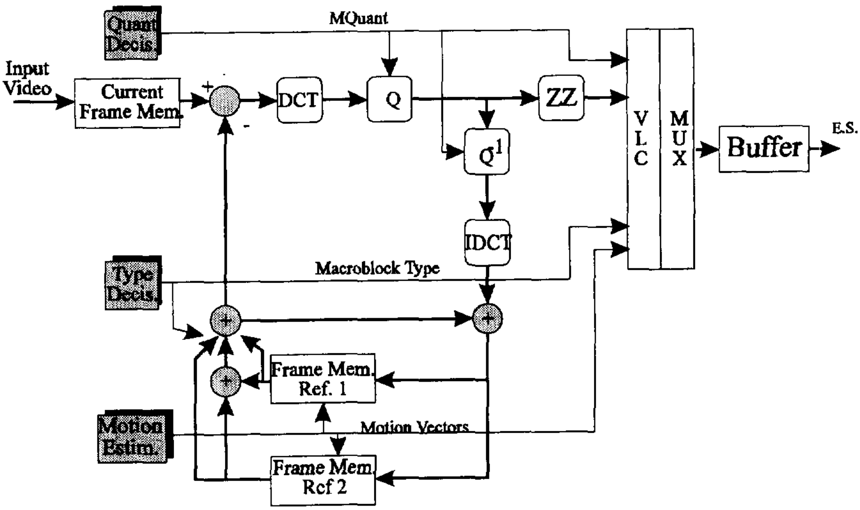


Figure 2.9: Schematic block diagram of an MPEG encoder

An MPEG encoder produces a legal MPEG bit stream that can be decoded by any MPEG decoder supporting the same profile and level. The freedom of implementation of an encoder resides essentially in three modules: the motion estimator, the motion compensation type decision and the quantization decision (adaptive quantization and rate control). A schematic diagram of an MPEG encoder can thus be as given in Figure 2.9, leaving the three modules as an abstraction because they may take a variety of signals as input.

2.3.8 Interlaced Input Format

To handle interlaced video sequences some extensions have been made to the non-scalable syntax when compared to the MPEG-1 syntax. Within MPEG two different approaches to interlaced video are possible. On the one hand, each single field of video can be considered an MPEG-picture yielding *field pictures*. Each picture can now be motion compensated and DCT encoded separately. The redundancy between the two fields can be exploited by predicting the second field from the first.
On the other hand there is the more established way of handling the interlaced video. In this method two fields are merged into one *frame picture*. The odd lines of the picture originate from one field, the even lines originate from the other field. It should however be noted that there is a sampling time difference between the two fields such that, in the case of motion, special measures will have to be taken.
For the spatial redundancy reduction of frame pictures, i.e. the intra picture coding, two different modes can be identified; intra-frame coding and intra-field coding. For macroblocks where there is little or no motion, frame-based coding will be performed as the adjacent lines of a frame are more correlated than the adjacent lines of a field. However, in the case of large horizontal motion, better results will be obtained by performing field-based coding. The internal organization of a macroblock is therefore distinctly different for intra-frame and intra-field coding, as is depicted for the luminance blocks in Figure 2.10 . A decision for the type of coding to be carried out has to be made for each macroblock and this information must be added to the syntax at the *Macroblock* layer.

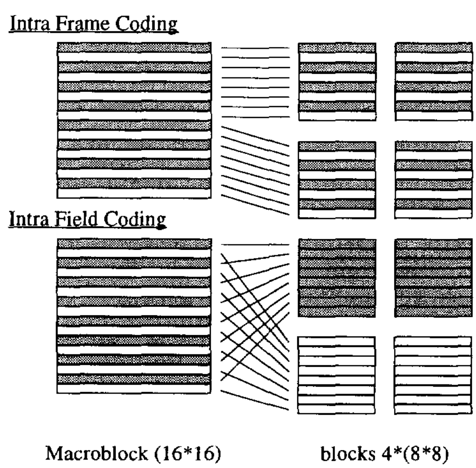


Figure 2.10: Macroblock organization for Intra frame versus intra field coding in frame pictures for interlaced video

With respect to the temporal redundancy reduction, again two different modes can be distinguished: inter-field and inter-frame motion compensation. In frame predicted macroblocks, there is one vector per macroblock per prediction direction. Vectors measure displacement on a frame sampling grid. Therefore an odd-valued vertical displacement causes a prediction from the field of opposite parity. Vertical half pixel values are interpolated between samples from fields of opposite parity. In field prediction macroblocks there are two vectors per macroblock per prediction direction; one for each field. Each field can be referenced to the corresponding field of the reference frame or to the field of opposite parity. The field vectors indicate the motion on a field sampling grid. The decision for the motion compensation mode can be made for each macroblock separately; the mode will be added to the *Macroblock* header.

2.3.9 Model Decoder for Buffer Control

The different frame types in the coded stream each require a different amount of bits to encode. In addition, the video may vary in complexity with time and an encoder may wish to spend more bits to one part of a sequence than another. For constant bit rate (CBR) coding, varying the number of bits allocated to each picture requires that the decoders have an input buffer to store the bits not needed to decode the immediate picture. The extent to which an encoder can vary the number of bits allocated to each picture depends on the size of this decoder buffer. Encoders need to know the size of the decoder buffer in order to determine to what extent they can vary the distribution of coding bits among the pictures in the sequence. The profiles and levels prescribe a buffer size such that all decoders of a certain profile and level will have a minimum buffer size.

The buffer size alone is not enough for guaranteed decodability. A buffer fullness mismatch of the decoder's input buffer may eventually cause a buffer overflow or underflow. The encoder should know the characteristics of the decoder with respect to the buffer operations, i.e. the encoder must know at what instances the decoder will take data from the buffer. Likewise, given a limited input buffer, the decoder should know how full its buffer must be before starting to decode. As the operation of every decoder is different, and it is not the objective of the standard to prescribe the timing in the hardware itself, a theoretical model decoder has been defined. This model decoder has an input buffer of which the encoder maintains a verifier, called the video buffering verifier (VBV), to periodically specify the buffer fullness in the bit stream. This fullness is specified in terms of the time required to fill the decoder buffer from empty to its current level, i.e. the VBV-delay.

The mechanism of the VBV is as follows: The coded bit stream is assumed to enter the decoder's input buffer at a constant rate. At regular intervals, set by the picture rate, the picture decoder instantaneously removes all the bits for the next picture from the input buffer. If there are too few bits in the input buffer, i.e. all the bits for the next picture have not been received, then the input buffer underflows and there

is an underflow error. If during the time between picture starts, the capacity of the input buffer is exceeded, then there is an overflow error.
The VBV is examined at picture intervals; At least one coded picture has to be present in the buffer and immediately after removing the above data, the buffer occupancy must be less than: B-X bits, where X=R/P, R=bit rate, P=picture rate.

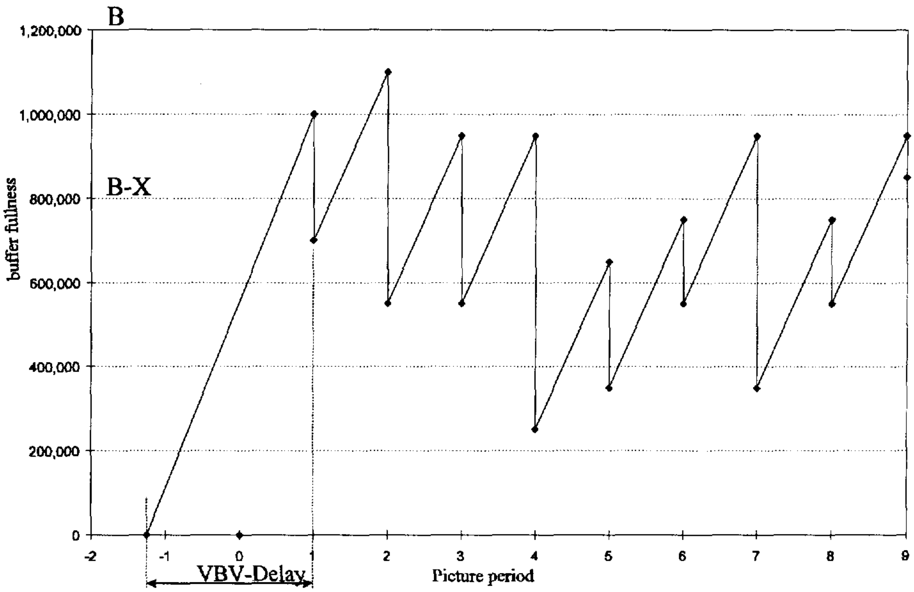


Figure 2.11: VBV Buffer occupancy

Figure 2.11 shows this VBV buffer fullness as a function of time; the buffer is constantly filled at the rate X and all the data needed to decode one picture from the buffer is instantaneously extracted when it is needed. The solid staircase line of Figure 2.12 shows the cumulative consumption of bits (the "bit-demand" curve) by the VBV, and thus by the model decoder, of a typical sequence.
Assuming a fixed buffer size, there is a time window during which a particular data element can be accessed by the VBV. Accessing data before this time is impossible as it has not been delivered by the channel. If it is not accessed within the time window the buffer will overflow. The availability of bits to the VBV is shown graphically in Figure 2.12 as a bit-supply region. In order to ensure correct operation of the decoder, the VBV requires that the bit-demand curve remains within the bit-supply region.
To ensure that the buffer neither underflows nor overflows it is necessary to partially fill the input buffer for a time period specified by VBV-delay before beginning to decode the pictures. After this delay the input buffer will be sufficiently full for normal decoding operations to begin. The VBV specifications guarantee that thereafter, while normal play is continued at the nominal data rate, the buffer will operate correctly. The light gray demand curves of Figure 2.12 show

that too long or too short a VBV-Delay will cause overflow or underflow to occur respectively, even though the beginning seems to work without a problem.

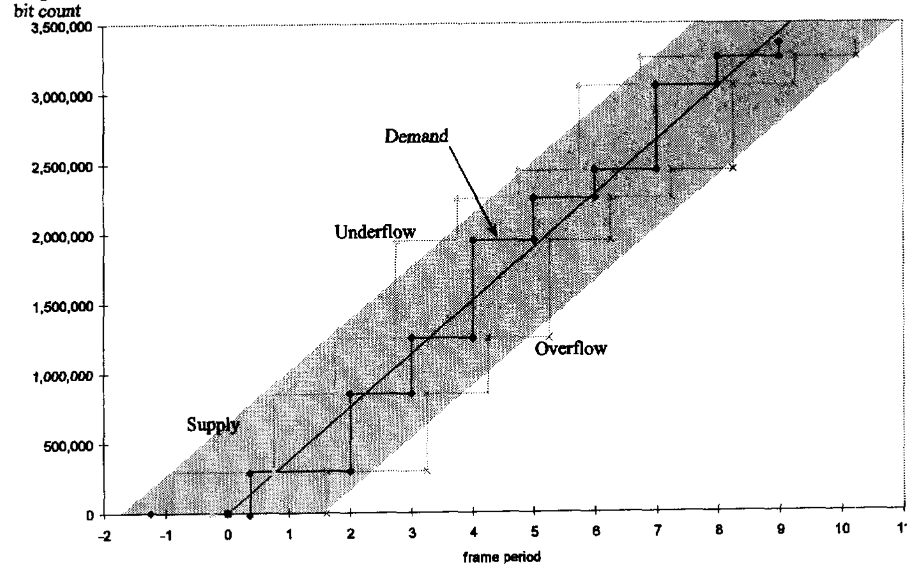


Figure 2.12: VBV bit demand and supply curve

Practical decoders differ from this model in several ways. They may not remove all the bits required to decode a picture from the input buffer instantaneously, they may not be able to control the start of decoding very precisely as required by the buffer fullness parameter in the picture header, and they take a finite time to remove the bits of one picture from the buffer. These differences depend on the particular method of implementation. Nevertheless practical implementations must ensure that they can decode the bit stream defined by this model. In many cases this will be achieved by using an Input Buffer that is larger than the minimum required, and by using a decoding delay that is larger than the value derived from the VBV delay.

2.3.10 Scalability

The scalable syntax is an additional feature developed to fulfill the requirements of specific applications. As this scalability may be important to the DART in certain cases, we will briefly present the main features.

The key property of a scalable bit stream is that a decoder can neglect part of the information in the bit stream, and still decode a useful picture. This is achieved by structuring the bit stream in two or more hierarchical scales, starting from a stand alone base scale and adding a number of enhancement scales. Some applications require the different scales to define different resolution hierarchies, others require the different scales to define different levels of quality. The base scale can use the non-scalable syntax or it can even be conform the MPEG-1 syntax [MPEG2V].

In addition to the benefit of multiple resolution or quality levels, all scalability approaches provide a means to enhance resilience to transmission errors by transmitting the most important lower scale over a channel with a better error performance. Four types of scalability are available in the standard: Spatial, SNR, Temporal and Data partitioning.

- ***Spatial***

Spatial scalability is a tool intended for use in video applications involving a hierarchy of resolutions (SDTV to HDTV) where a minimum of two spatial resolution scales are necessary. The lower hierarchical scale is the individually decodable lower resolution. The higher scales form a higher resolution enhancement on top of the spatially interpolated lower scales.

- ***SNR***

SNR scalability is a tool intended for use in video applications involving a hierarchy of qualities. SNR scalability is comparable to Spatial scalability, with the difference that instead of a resolution step from one scale to the next there is a quality step from one scale to the next; in SNR scalability all scales have the same spatial resolution.

- ***Temporal***

Temporal scalability is intended for future applications in which migration to higher frame rates is required. The lowest scale would use the non-scalable syntax with its own temporal predictions. The higher temporal enhancement scale is coded with temporal predictions with respect to the lower scale.

- ***Data partitioning***

Data partitioning is a tool intended to provide a low complexity means of enhanced error resilience. In the data partitioning the lower scale contains the more critical parts of the bit stream and the higher scale contains less critical parts. As is the case for spatial and SNR scalability, the lower scale can be transmitted in a channel with better error resilience. Unlike the other scalability solutions the lower scale is not necessarily independently decodable.

2.4 MPEG System

In the previous section we have discussed the video stream syntax and the possible switches available to the encoder. The structure of the video bit stream to be recorded by the DART is thus identified such that the implications of recording the video signal can be studied. The physical interfacing of the received signal to the recorder is however performed on the system layer. Besides the MPEG video syntax it is therefore important to study the MPEG system syntax [MPEG2S] and its implications to the subject of recording.

2.4.1 Multiplex Solution

The system component of the MPEG standard supports the combination of coded video and audio in a single multiplexed stream. Figure 2.13 shows a typical system configuration, where at the encoder side multiple video and audio signals are multiplexed into a single stream. At the playback side the de-multiplexer directs the individual streams to the appropriate decoders.

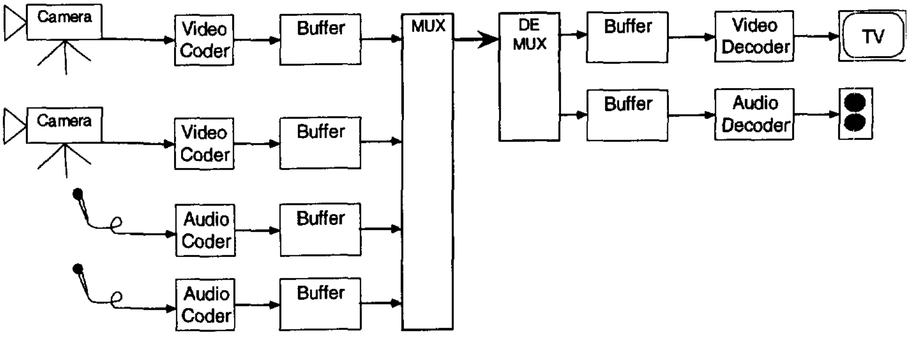


Figure 2.13: Typical system configuration

The system layer supports the following two basic functions:

1. The interleaving of multiple compressed streams into a single stream
2. The end-to-end timing of the individual streams

The system layer is a byte oriented time division multiplexer. The multiplex is packet-based where the packet headers can contain information to support the timing and buffer control functions. The packet payloads stem from the elementary video, audio or data streams. The system standard has two forms: The *Transport stream* and the *Program stream*.

The *Program stream* is analogous to the MPEG-1 system specification [MPEG1S]. It results from combining the Packetized Elementary Streams (PES) of one of more streams with a common time base into a single stream. The *Program stream* is designed for use in a relatively error free environment and is in particularly suitable for applications which involve software processing of system information such as interactive multimedia applications. *Program stream* packets may be of variable and relatively great length.

The *Transport stream* combines one or more programs with independent time bases into a single stream. PES packets of different streams, that together form a program, can share a common time base. The *Transport stream* is designed for use in environments where errors are likely, such as storage or transmission in lossy or noisy media. The *Transport stream* packets are 188 bytes long.

Transport streams are of particular interest to high bandwidth broadcasting systems, where multiple programs from different broadcasters are multiplexed. The Transport stream is also very well suited for transmission on IBCN; the fixed length packets can easily be mapped on four ATM cells. Given the commonality between the applications targeted by the DART recorder and those intended to use the Transport stream, we will limit our discussion on the MPEG system to the Transport stream.

2.4.2 Multiplex Syntax

The Transport stream is divided into two sub-layers, one for multiplex wide operations (the transport packet layer) and one for stream-specific operations (the PES layer). Figure 2.14 shows the sub-layer structure.

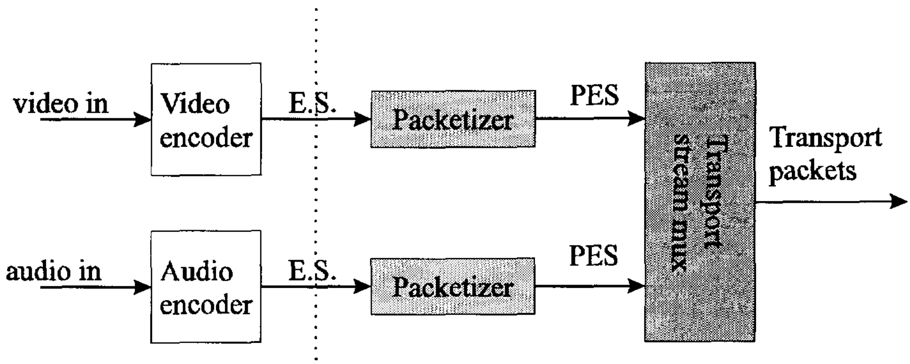


Figure 2.14: Transport stream sub-layers

- ***Packetized Elementary Streams***

The PES packets contain coded bytes from only one elementary video or audio stream. A PES packet consists of a PES packet header followed by the packet payload. Each packet starts with a 32 bit *start_code* like the start codes defined at the beginning of each layer of the video stream. The *start_code* contains an ID code which identifies the stream to which the packet belongs. The PES header may also contain decoding or presentation time stamps such that the timing of the individual streams is controlled.
The PES payload contains a variable number of contiguous bytes from one elementary stream. The payload is obtained by subdividing the elementary streams in packets.

- ***Transport Stream Packets***

The transport packet layer coordinates the data retrieval of the channel, the adjustment of the clocks and the management of the buffers. Transport stream packet headers contain information which specify the time at which each byte is intended to enter the transport decoder.
PES packets make up the payloads of the Transport stream packets. PES packets are much larger than the Transport stream packets such that one PES packet will span several Transport stream packets. The PES and the Transport stream packet do not form a strict layered structure; i.e. the PES boundaries do not have to be aligned

with the Transport stream packet boundaries. However, a Transport stream packet will never carry data of more than one elementary stream.
The Transport stream packets commence with a 4 byte prefix which contains a 13 bit packet ID (PID). The PID identifies the contents of the data contained in the transport packet. The PID thus forms the key to the multiplexing function of the system.

2.4.3 Timing Aspects

In practical applications, subtle problems of synchronization occur. In these applications the encoder should generate a bit stream at a specified bit rate. The decoder will display the decoded pictures at their specified frame rate. If the display clock is not locked to the channel data rate, and this is typically the case, then any mismatch between the encoder clock and the display clock will eventually cause a buffer overflow or underflow problem. In order to alleviate this timing problem, encoder clock information needs to be transmitted along with the compressed data.
Figure 2.15 illustrates the prototypical de-multiplexing and decoding systems. The Transport stream is de-multiplexed in the system decoder. After de-multiplexing the appropriate elementary streams are passed to the decoders. The system decoder also extracts the timing information from the Transport stream.

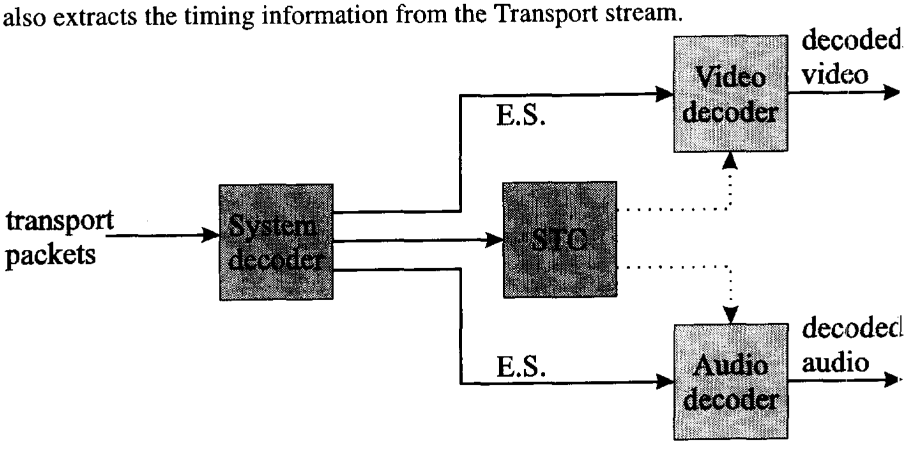


Figure 2.15: Prototypical de-multiplexer

In order to provide a formalism for the timing and the buffering relationship a System Target Decoder (STD) is used. The STD provides a model to specify the timing of the elementary streams. In this model the end-to-end delay from the signal input to the signal output is constant. This delay is the sum of the encoding, buffering, multiplexing, communication or storage, de-multiplexing, decoder buffering, decoding and presentation. The timing model assumes that the inter-picture interval and audio sample rate are the same at the decoder as at the encoder. Actual decoders will not follow this model. It is the responsibility of the decoder that the deviations from the model are compensated for.

All timing is defined in terms of a single system time clock (STC). In the transport stream the system clock frequency is constrained to have an exactly specified ratio to the audio and video clocks at all times. The STC is synchronized by means of system clock references (SCR) included in the transport stream.
For the synchronized presentation of multiple streams with a common time base the Transport stream contains presentation time stamps (PTS). The presentation time stamps indicate the presentation time of the presentation units packaged in the Transport stream. In case of video, a presentation unit is a picture; in case of audio a presentation unit is an audio block. The individual video and audio streams are thus synchronized by locking the decoding and presentation time to the time stamps.

2.4.4 Trick Mode Tools

Within the PES packets of the system *Transport stream* syntax an optional trick mode byte is available. This byte provides the tool to signal relevant parameters from the digital storage media (DSM) to the decoder. In this section we will evaluate the usability of this signaling in the DART.
The first three bits of the trick mode byte indicate the current *trick_mode_type*. The following are the possibilities:

- Fast forward (FF)
- Fast reverse (FR)
- Freeze frame (Z)
- Slow forward (SF)
- Slow reverse (SR)

Depending on the trick mode type the remaining bits are used for some additional information. For Fast playback (FF and FR) a *field-id* indicates which field(s) are to be displayed, an *intra_slice_flag* indicates that there may be missing macroblocks between coded slices and a *frequency_truncation* indicates that a restricted set of DCT coefficients has been used. For slow motion (both SF and SR) a *field_rep_cntrl* indicates the number of times that a progressive picture should be displayed.
For the slow motion and freeze frame application the signaling is the most important part. The picture itself can be built up from the normal stream. For the fast playback it is, however, not defined what the fast playback bit stream will be and this stream must consequently be built by the DSM adaptation interface.
The MPEG video standard makes some suggestions how the decoder should react to the trick mode signals. When toggling from normal play to trick modes or visa versa the decoder is suggested to remove all unused data from its input buffer. During trick mode decoding the decoder is suggested to ignore the buffer fullness parameter (VBV) and temporal reference value. Furthermore it is suggested to decode one picture at a time and display it until the next picture is decoded.
It should be noted that these suggestions do not form part of the standard and consequently it is unclear to what extent "generic" decoders will follow them. For

our purpose we cannot assume that these trick mode features will be supported by the decoders such that little advantage will be gained from the trick mode tools.

2.5 System Aspects of MPEG Recording

2.5.1 Interfacing Point

The expected wide spread usage of the transport stream the typical DART applications yields to the conclusion that in our problem of MPEG recording the actual point of interfacing is the transport stream. What stream actually should be recorded by the DART system remains to be decided at this point. The following bit streams are the identified candidates:

- transport stream packets
- separate packtized elementary stream packets (PES)
- separate elementary streams (video and audio)

Given the features of the MPEG Transport stream with respect to multiplexing, timing control, buffer control and error resilience, it is clear that recording the Transport stream packets has the clear advantage. Recording PES streams has as disadvantage that only one E.S. can be recorded as any defined relation between the elementary streams will be lost. Furthermore, given the variable length of the PES packets there will be a reduced error resilience. Recording separate elementary streams has as additional problem that all the timing information will be lost.
Recording Transport stream packets has the advantage that the DART multiplex is implicitly defined. It is possible to record one program from the Transport stream but is also possible to record multiple programs at the same time. The number of streams included in the multiplexed stream is in principle of no concern. There is thus no need for the DART project to define a multiplex.
In this thesis, the prime application of concern is, however, the recording of a single program, with one video signal and possibly several audio and data signals.

2.5.2 Tape Mapping: Sync-blocks versus Transport Packets

For the verification model hardware we have chosen not to make any use of the Transport stream. There are several motivations for this decision:

- At the time of the definition of the verification model the MPEG system standard was still in its definition phase. As such, only preliminary versions were available. In particular the Transport stream was only define well after the start of the project.
- No hardware was (or still is) available for *Transport stream* encoding or decoding (status early 1995). The added complexity of implementing the *Transport stream* is significant.
- The main point of interest, i.e. video fast playback, is not influenced by the particular implementation of the MPEG system.

For the DART verification model the available bit machine uses a sync-block based communication which stems from the DVT project. Sync-blocks are essentially fixed length sections on tape with the length 128 or 77 bytes. For a more thorough discussion on the sync-blocks we refer to the discussion on the recording system (Section 3.2.2). At this time the important point to note is that the sync-block based protocol, used throughout this thesis, can be viewed as a low level packet system. Each sync-block forms a packet with a packet number; sync-blocks can therefore be considered as DART packets. Static multiplexing can be performed based on packet number ranges, i.e. all packets within a certain range can belong to a certain stream.

Although the ultimately desired "system" is the Transport stream, the sync-blocks are a simplified intermediate used for the experimental evaluation. When only one video stream is recorded then, for all practical purposes, there is no fundamental difference between the usage of the transport stream and the usage of our own sync-blocks. At the relevant points in this thesis we will indicate what the significance is of the substitution of Transport stream packets by the somewhat shorter sync-blocks.

2.5.3 Implications for Hardware Validation

The sync-blocks contain a header with little more than a packet number (sync-block number). There is no timing information included in the bit stream. This is an important difference with the Transport stream for the hardware validation. Because we lack a timing mechanism an alternate approach to locking the frame clocks to each other will have to be taken.

Figure 2.16 shows the typical situation under consideration, where only one video stream is recorded. The audio streams are not considered at this time. A 50 Hz video ($C_{dis,r}$) sequence is encoded and the resulting video bit stream is buffered to convert it into a constant bit rate. The constant rate video is packetized and the adaptation interface puts the packets on the channel (the bit machine). On the playback side (receiver) the adaptation interface recovers the packets which are depacketized and the resulting video bit stream is entered into the decoder buffer. The decoder takes frames from the buffer at a rate governed by its display rate ($C_{dis,p}$).

Given a fixed amount of bits per packet the rates $C_{bit,r}$ and $C_{packet,r}$ are locked to each other. The same applies to $C_{packet,p}$ and $C_{bit,p}$. For channels in general, $C_{packet,r}$ and $C_{packet,p}$ are locked to each other.

In a system that employs a Transport stream the display frequencies $C_{dis,r}$ and $C_{dis,p}$ would be locked to each other by the insertion of presentation time stamps in the bit stream. With this lock a constant end-to-end synchronization would be assured and we would have the guarantee that the decoder buffer will never overflow or underflow, as a result of clock discrepancies.

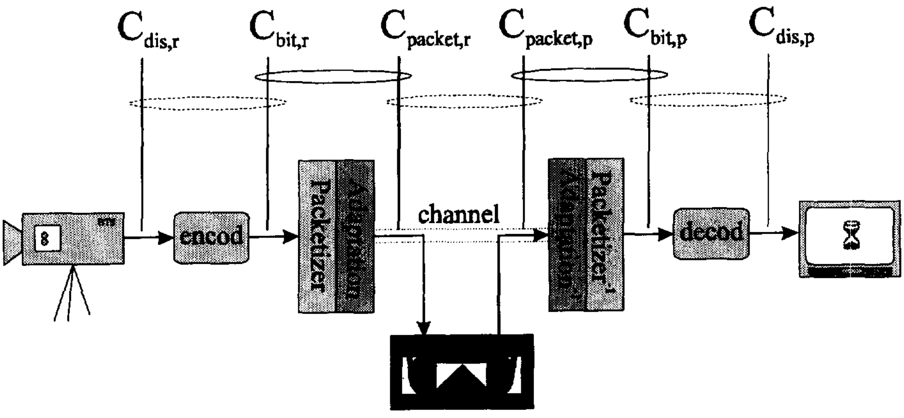


Figure 2.16: Verification timing

In our case, where we do not have the possibility to transmit time stamps, there is only one possibility to lock the two display frequencies $C_{dis,r}$ and $C_{dis,p}$ to each other, i.e. by locking $C_{dis,r}$ to $C_{bit,r}$ and $C_{dis,p}$ to $C_{bit,p}$. The lock of the display clock to the bit rate can be achieved by choosing a fixed ratio between the two.

We can thus conclude that the choice not to transmit any timing information in the sync-blocks implies that the clocks of the whole system, in both recording and playback, should be locked to each other by choosing fixed clock ratios.

3. Tape formatting for fast playback

3.1 Introduction

The way in which recorded data is mapped onto tape is called the tape format. The specific formatting of the MPEG video data on tape has a significant effect on the performance of fast playback. Depending on the tape format, different sections from the recorded bit stream will be read. In this chapter attention is paid to the design of appropriate tape formats to support fast playback. Two forms of realizing fast playback are considered, namely rudimentary fast playback and fast playback using a dedicated stream.

The design of the tape formats requires a thorough understanding of the process of reading data by of the helical scan recorder when the tape is played back at a higher speed. In Section 3.2 the helical scan recording principle is presented and a model for the fast playback signal is developed. The helical-scan recorder is evaluated from a system point of view: the model therefore describes the external behavior relevant to MPEG recording. Based on this model the tape formats for two different fast playback methods are developed. In Section 3.4 the formatting for *rudimentary fast playback* is studied; no extra information is recorded in addition to the normal play stream. In Section 3.5 the formatting required for the playback of *dedicated fast playback* streams is developed. This chapter concentrates on the tape format design for the two methods. Their performance is evaluated in Chapters 4 and 5 respectively.

3.2 Helical Scan Recorder

3.2.1 Helical Scan Recording Principle

The mechadeck at the heart of the DART bit machine is a helical-scan recording system. Helical scan recording, which is depicted in Figure 3.1 [Suga88,With92b,With93a], forms the basis of all analog consumer video recorders. It also is an important technique for digital recording. In helical-scan recording the magnetic heads are mounted on a rotary head wheel inside a cylindrical drum, which has the tape wrapped helically around it. In principle the wrap angle of the tape around the drum is 180°, though other wrap angles are possible. The tracks written on tape have a small angle with respect to the tape travel direction and are consequently gradually crossing the tape width. The entire configuration of the drum, the head wheel and the heads is called the scanner.

In helical scan recording systems, a high recording density is achieved by both high linear density (short bits) and high track density (narrow bits). High track density is achieved by using a small track pitch and azimuth recording: track pitch is the distance between the centers of two adjacent tracks, azimuth is the angle between the actual gapline of the magnetic head and the direction perpendicular to the track length. This azimuth angle determines the orientation of the bit written to tape. In azimuth recording two heads with different azimuth are used and the bits of two adjacent tracks thus have a different orientation as shown in Figure 3.2. Azimuth recording allows tracks to be written to tape without the usage of a guard band between the tracks. When head A cross-tracks into track B, the resultant difference in angle between the A head and the B track is such that there is a significant loss of SNR of the signal from track B. Therefore, head A will only successfully read the signal from track A and head B will only read the signal from track B.

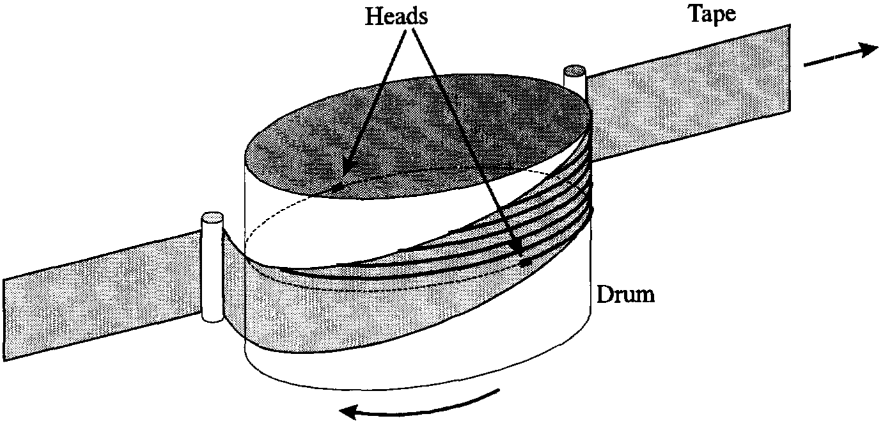


Figure 3.1: Helical-scan tape recorder with two heads

Referring to the basic system of Figure 3.1, the most straightforward solution for azimuth recording is one where the two heads are at opposite sides of the scanner and have opposite azimuth. When a track of azimuth A (0° on the head wheel) has been written by head A, then head B (at 180° on the head wheel) takes over and starts writing the next track with azimuth B.

It should be noted that many different solutions to azimuth recording exist, depending on the wrap angle and the number and type of heads used. An equivalent solution to the above is one where the head wheel contains one head *pair*, consisting of two heads of opposite azimuth mounted directly next to each other. In such a system two tracks of opposite azimuth (sometimes called a *double track*) are written simultaneously and this is done on every revolution of 360° of the head wheel. A recorder with a double transfer rate is obtained by mounting a second head pair at 180° on the head wheel and doubling the tape transport speed.

An important remaining point is that of the tracking and the locking of the frequency and the phase of the scanner to the tape. This is necessary to assure machine to machine interchangeability. In all systems, the phase of the scanner will be locked to the tape, such that the head will be positioned above a track. In many existing helical scan systems the tracking is performed using a control signal from a longitudinal track on the tape. In the newest systems the tracking is performed using embedded pilot signals which are recorded along with the video signal.

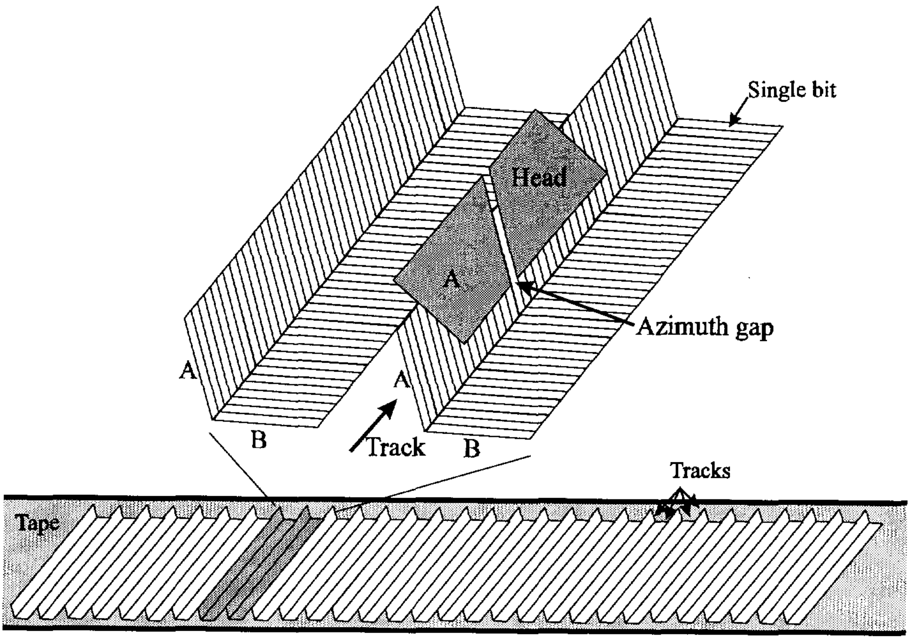


Figure 3.2: Azimuth recording track pattern

3.2.2 Black Box Interfacing

In the concept of the DART, the recording of MPEG video is only one of the many features of the system. The bit machine at the core of the system is thus specified to meet the needs of many different applications and the MPEG recording requirements have moderate influence on the bit machine. The bit machine is therefore pre-defined and it is important to analyze the external black box behavior at the point of interfacing for the adaptation interface.

A conceptual block diagram of the bit machine is given in Figure 3.3, where the mechadeck (symbolized by a cassette), the channel electronics and the error correction electronics (Erco) can be distinguished. The adaptation layer will interface with the Erco unit which has two levels of parity. The inner (horizontal) parity is mainly meant to correct random errors and operates on sync-blocks, i.e. the data units of which a track is composed and on which the system can synchronize. A sync-block is the smallest data unit that can be individually passed or rejected during playback. The raw data size of the sync-block is system dependent; both 77 Bytes/sync. and 128 Bytes/sync. are possible. The outer (vertical) parity works across sync-blocks of the same tracks; for the 77 Bytes/sync system there are 138 sync-blocks per track. It is meant to correct burst errors and will only be effective if sufficient sync-blocks of the same track are recovered.

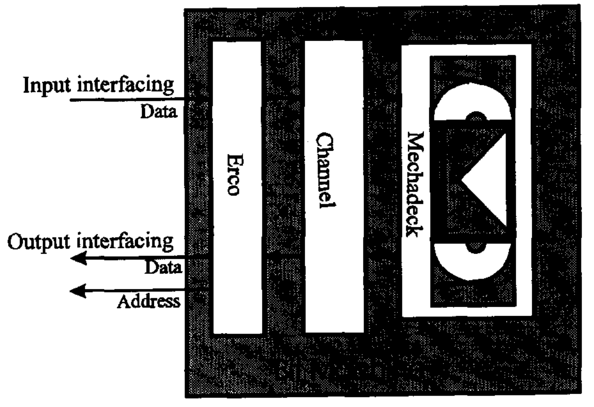


Figure 3.3: Bit machine at the heart of DART

The entire communication of the bit machine with the outside world is sync-block oriented. The sync-blocks from a number of tracks are grouped in a Data Frame Structure (DFS). Within a DFS the sync-blocks have a unique address which the bit machine adds to the sync-block header during recording and which is passed along with the sync-block data to the output interface on playback.
Both the sync-blocks and the DFS grouping are historically determined. In our work we will use these concepts as a basis to describe the black box behavior of the system. Our adherence to the existing system is intended to ease the transition from the paper design to the hardware demonstrator. The evaluated approaches will however not require the actual usage of "sync-blocks" and DFS's and will be more generally applicable.

3.2.3 Different Available Bit Machines

The basic bit machine has a usable bit rate of 25Mb/s. This bit rate is achieved with two heads of opposite azimuth. The heads can either be a head pair, mounted next to each other on the head wheel or they can be mounted on opposite sides (180°). For normal playback the two solutions are equivalent; for fast playback there are however some essential differences. Within the DART project the configuration with a single head pair is favored such that a track pair is written to tape in a single pass.
Other configurations of the bit machine are also possible. An important configuration is the bit machine which supports a usable bit rate of 50Mb/s by using four heads and doubling the tape speed. The head wheel then contains two head pairs at opposite sides (180°). In principle this is not the targeted machine for the DART. This machine can also be operated at 25Mb/s by discarding one head pair.

Then this machine is identical to the 25Mb/s version with an additional option of a higher transfer rate.
Both the 50Mb/s and the 25Mb/s machine can support an integer fraction of the maximal bit rate. The main parameters of the mechadeck like scanner speed and actual scanner to tape bit rate are kept unchanged, but the tape speed is reduced and the heads write data to tape only a fraction of the time. For example a 12.5Mb/s operation of the 25Mb/s machine can be defined by halving the tape speed and halving the usage of the head pair; the head wheel will rotate 720° between the beginning of the scans. Although in principle other bit rates than 12.5 Mb/s are possible, the practical implementation is limited by the servo systems.
For our purposes, three different core machines are important: the 50Mb/s machine with two head pairs, the 25Mb/s machine with one head pair and the 12.5Mb/s machine with one head pair which writes data on every other rotation. We will identify these machines or different operation modes as M1, M2 and M4 respectively.
For the M1 machine the DFS has the size of 48 tracks. For the lower rate bit machines the DFS rate is kept constant (12.5Hz). Consequently the DFS contains 24 single tracks for the M2 machine and it contains 12 single tracks for the M4 machine. For all the above systems the amount of usable data in a track (L_T) is about L_T=85 kbit. This usable data does not span the entire track; part of the track is used for other data sections and for the parity section.

3.2.4 Read Mask in Fast Forward

Figure 3.4 symbolically shows the tracks on the tape where the tracks are drawn perpendicular to the tape travel direction. In practical systems this is not the case but this abstraction eases the visualization of the problem without loss of generality.
In normal playback, a reading head pair traces along recorded tracks because during playback the tape speed is equal to the recording tape speed. The trace follows the tracks on tape as a result of the sum of the tape speed and the scanner speed. The scanner speed is significantly larger than the tape speed.

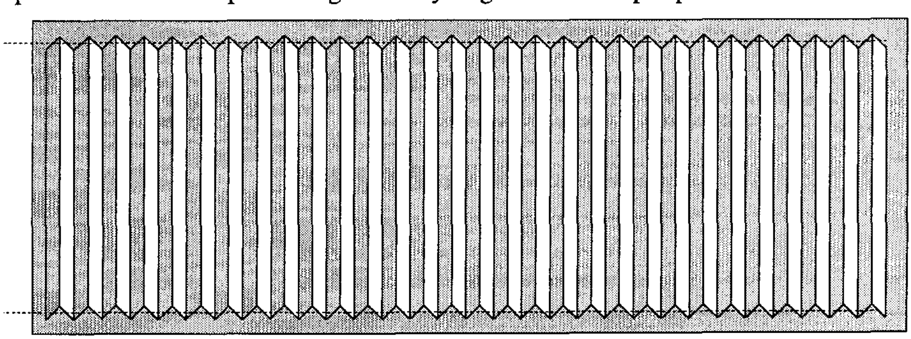


Figure 3.4: Symbolical drawing of tracks on tape

One of the main differences between storage and transmission is the possibility of different playback speeds. The operation at a speed different from the normal playback speed dramatically influences the amount of information that can be recovered from the individual tracks.

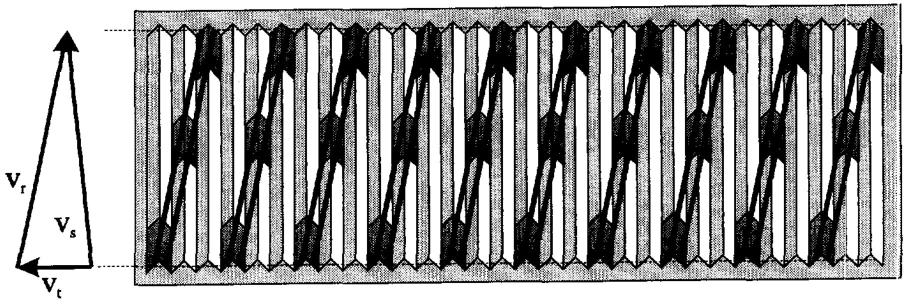


Figure 3.5: Recovered sections (read mask) in fast forward playback mode (n=3.0) for the M1 bit machine

In the fast forward mode, the scanner speed (v_s) is kept constant and the tape speed is increased. Consequently the head pair will trace across several tracks, such that only fragments of the tracks can be recovered. In Figure 3.5 the sections that can be read by the M1 machine, for the particular speed-up of three times the normal speed, are shaded. The vector of the heads over the tracks, the trace v_r, is defined by the combined tape speed (v_t) and the scanner speed (v_s). In Figure 3.5 these traces of the head pairs are drawn with a double line, where each line represents the center of a head. When a trace has crossed the tape, the other head pair at 180° will take over and start a trace at a position that is determined by the angle of the scanner vector v_s. Figure 3.6 shows the situation for the fast reverse case with a speedup of three times the normal play speed.

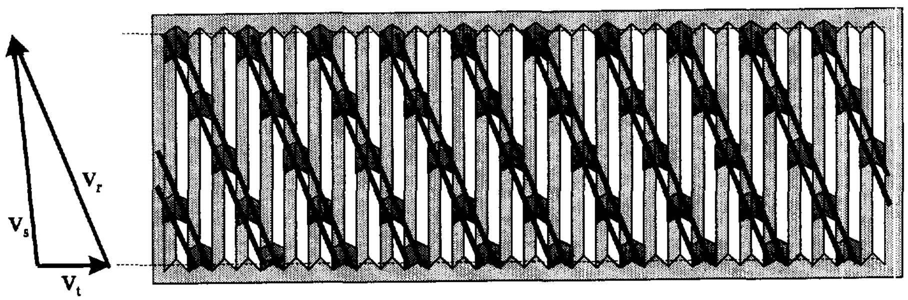


Figure 3.6: Recovered sections (read mask) in fast reverse playback mode (n=-3.0) for the M1 bit machine

Note that there is still one degree of freedom left: the scanner to tape phase. For a different scanner to tape phase, the phase of the beginning of the traces with respect to the tracks will be different and therefore the pattern of successfully read sections, i.e. the read mask, will be different.

3.2.5 Read Mask for Different Mechadecks

The nature of the extracted signal depends on the specific properties of the recorder used. In particular, the head configuration and the mode of operation during normal play will yield different results.

The situation for the M1 machine is shown in Figure 3.5. A double track is written to tape in the time taken to rotate the scanner by 180°. During this time the tape has advanced exactly the right amount such that the alternate head pair can immediately start writing the next track pair directly next to the previous track pair. In normal playback the operation is identical and in fast playback the only change is that the tape speed is changed.

Figure 3.7 shows the situation for the M2 machine with a single head pair. As the wrap angle of the tape is still 180° a double track is still written in the time the scanner rotates 180°. The next track pair is however written by the same head pair after the scanner has rotated through the full 360° (M*180). In this time the tape has advanced exactly the right amount such that the next track pair is written adjacent to the previous track pair. The nominal tape speed is half of what it was for the M1 machine. As the tracks are still symbolically drawn perpendicular to the tape, the same track pattern is obtained as before. Note however that the physical tracks on the tape have a slightly different angle with respect to the tape travel direction than is the case for the M1 machine.

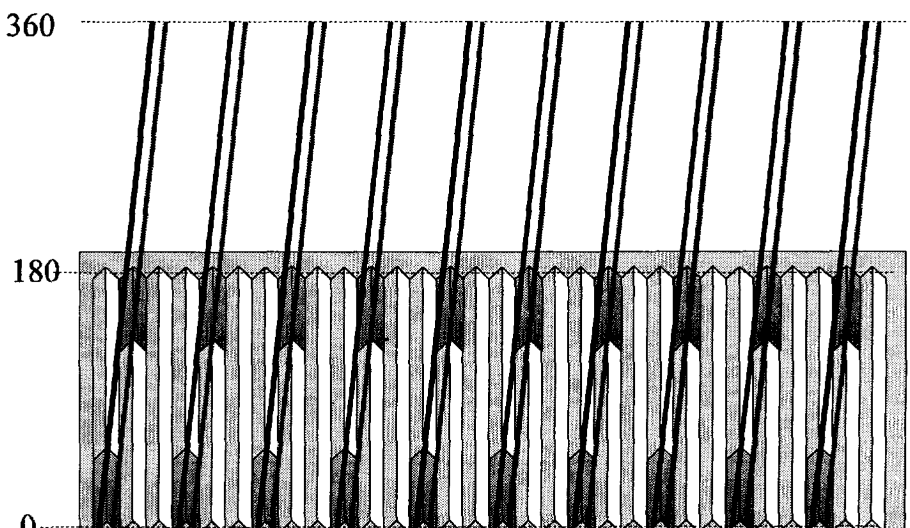


Figure 3.7: Read mask for the M2 machine (n=3.0)

In fast playback the track read pattern is quite different from the M1 machine. The resulting angle between a fast forward trace and the tracks on tape is smaller such that longer sections of data are read from a particular track at the cost of reducing the amount of sections from which data is read. The resulting track read pattern for the speedup of (n=3.0) at a particular phase is shown by the shaded sections of the figure.

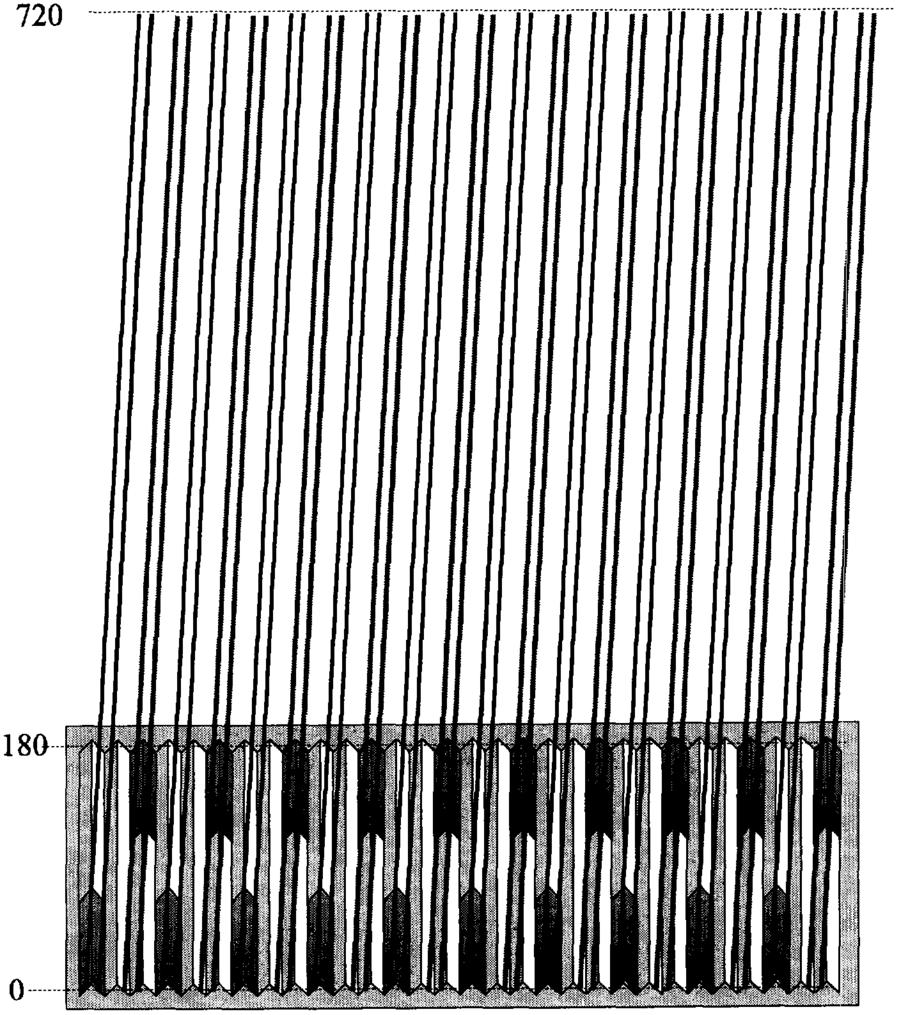


Figure 3.8: Read mask for the M4 machine (n=3.0)

The situation for the M4 machine is extrapolated in a similar manner. The scanner rotates through 720° between the writing of two consecutive track pair on tape. During fast forward, the angle between the trace and the track is consequently

reduced when compared to the M2 case, such that even longer sections of data are read.
With the M4 machine in fast playback, advantage can be taken from the fact that the head pair passes the tape an extra time between consecutive scans. We can thus define an enhanced trick playback operation, with an enhancement factor E=2, in which every occasion is used to extract data from the tape. The resulting track read pattern is shown by the shaded areas of Figure 3.8. In this figure the black lines represent the traces without the enhancement (E=1), the gray lines represent the added traces due to the enhancement.
We can conclude that during fast playback the read mask is a dependent upon the speedup factor, the scanner to tape phase and the bit machine used. For the different bit machines M1, M2 and M4 a point to note is that the individual bursts get longer for the lower bit rate operation.

3.2.6 Model for Fast Playback Read Mask

From the previous section we have a global impression of the fast playback read mask for the different bit machines. We will now complete the model by including the phase dependency as a parameter and by a rough estimate of the lengths of the individual burst. The resulting read mask model will be equally applicable to fast forward and fast reverse.
The model is based on the geometric representation of Section 3.2.4. Important parameters are: the speedup factor (n), the scanner to tape phase (p), and finally the Q-factor. We define the phase as the position of the beginning of a trace with respect to the tracks on tape, where a unit phase represents an offset by the track pitch.
Given an initial scanner phase (p) and the speedup (n), the model geometrically predicts where each of the heads will be aligned with the tracks of corresponding azimuth for the given recording system. For a speedup of n=3 and a phase p=0.0 this model thus corresponds with the figures of the previous sections.

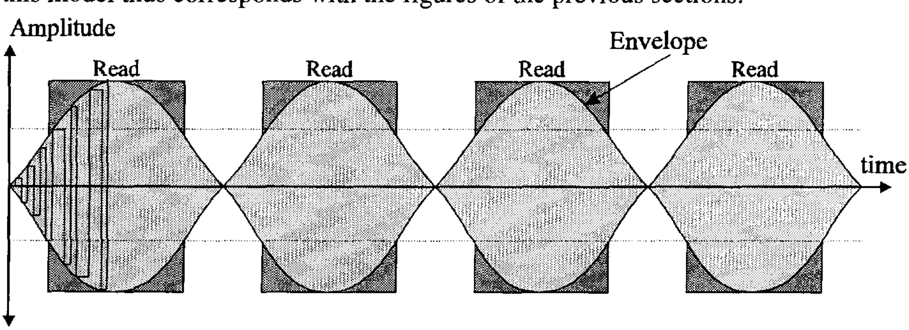


Figure 3.9: Recovered signal envelope

At the point where a head is positioned directly above the correct track, the recovered signal will be the strongest and the information can be read from the tape. When a head is above a track of the opposite azimuth, no information will be read at all. This implies that the recovered signal will be contained by an envelope as shown by Figure 3.9 . The signal will only be read error-free when the SNR is higher than a certain level. The nature of the envelope and the level of required SNR is very recorder specific.

To come to an abstraction of the size of data sections that can be recovered, the Q-factor ($0 \leq Q \leq 1$) is used. The Q-factor indicates what fraction of the recovered signal can be used, i.e. if Q=0.5 half of the recovered signal will be read error free. Implicitly this means that a usable signal will be read if more than half of the head is above a track of correct azimuth.

The average fraction of data read from tape (f_{READ}) depends on the tape speed and is defined as follows:

$$f_{READ} = E \cdot \frac{Q}{|n|} \qquad (0 \leq f_{READ} \leq 1) \tag{3-1}$$

where E, the enhancement factor, is E=1 for M1 and M2 and E=2 for M4.

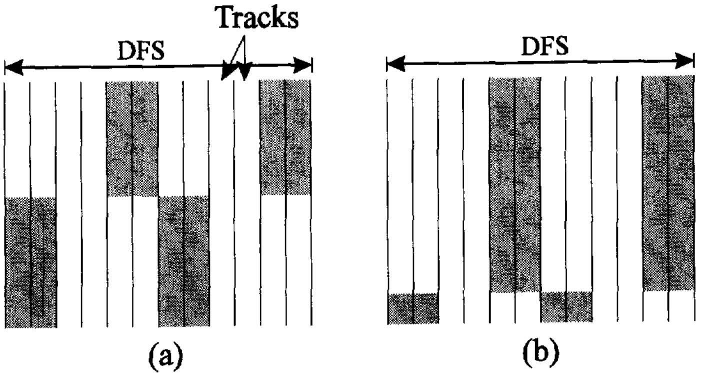


Figure 3.10: Read mask for (a) p=0.0 and (b) p=0.4 at n=3.0, Q=0.5

Given the geometrics of a trace during fast playback, where n-1 tracks are crossed horizontally while $M*L_T$ bits are crossed vertically, the maximum size of a contiguous burst (L_B) of data from a single track is given by:

$$L_B = \min(Q \cdot \frac{M \cdot L_T}{|n-1|}, L_T) \qquad (n \neq 1) \tag{3-2}$$

Depending on the scanner to tape phase the actual length of the tape sections read will usually be less than this maximum. The abstraction of the Q-factor can be used in conjunction with the known geometrics of the fast playback traces (e.g. Figure 3.5, Figure 3.7 and Figure 3.8) to predict the size of the data sections read as a function of the speedup n and the phase p. The result of this model for the M4 machine with n=3.0, Q=0.5 is shown in Figure 3.10.a for p=0.0 and in Figure

3.10.b for p=0.4. Only data from the shaded track sections can be read error free, i.e. the recovered sync-blocks are shaded.
To study which data sections are read for every possible phase, the phase-read diagram representation of Figure 3.11 is used. In this new representation the entire read mask of a DFS for a particular phase is transformed to single line by concatenating the tracks head-to-tail. As such, a single horizontal line of the diagram is the result of the model for a particular phase; vertically different scanner to tape phases are considered. Figure 3.10.a is transformed to the line with phase p=0.0 and Figure 3.10.b is transformed to the line with p=0.4 If a bit stream is packetized into sync-blocks and these are written to tape on a track by track basis, then the horizontal axis represents the packet train, where the dark areas (mask) in the diagram indicates which packets will be read for a particular phase. It can thus be seen that, if the phase is unknown, it is impossible to determine what data sections will be read.

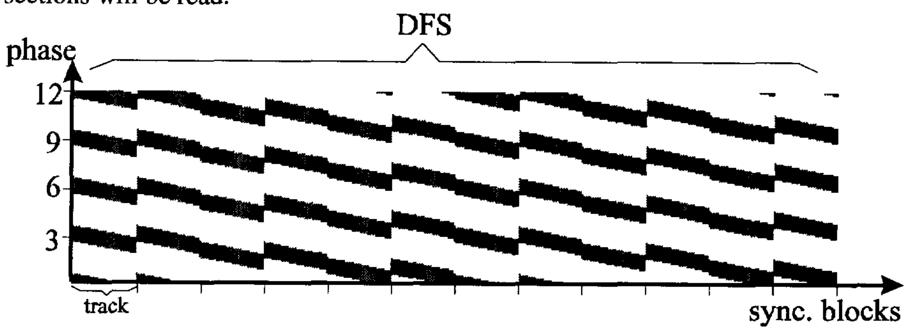


Figure 3.11: Phase read diagram (n=3.0, Q=0.5).

The read mask model is a black box description of the helical scan recorder during fast playback. To the application, the only matter of interest about the recorder during fast playback is what bit stream sections will be recovered. Most of the model simplifications with respect to the amount of data read, are captured by the Q-factor. Some finer geometric parameters like the exact head positioning on the scanner are neglected at this time. In Chapter 6 the model will be validated by comparing it to the data recovered from an actual system.

The model is applicable to systems with and without phase locking between the scanner and the tape. The entire Figure 3.11 must be used for a system without phase locking. For a system where phase locking on a track basis is possible, only the integer phases of the figure are relevant. In an even more advanced system, where during fast forward the phase can be controlled to read certain areas on the tape, only a single horizontal line of the figure would be relevant.

3.3 Black Box Adaptation

At this point we have enough knowledge to define the tape formatting functionality of the MPEG recording adaptation layer between the recorder and the MPEG bit stream. On one hand Section 3.2 provides us with a model that describes the black box behavior of the recorder during fast playback. On the other hand Chapter 2 describes the syntactic structure of the MPEG bit stream to be recorded. The objective now is to link these two *black-box* descriptions to define the adaptation layer shown in Figure 2.1. Two distinct approaches to fast playback can be identified: rudimentary fast playback and dedicated stream fast playback.

3.3.1 Rudimentary Fast Playback

The most simple approach is to use shreds of recovered data from the recorded normal bit stream to reconstruct a fast playback picture. The objective of this rudimentary fast playback approach therefore is to perform fast playback while no other data than the normal play bit stream will be recorded to tape. As far as the recording is concerned, the functionality of the adaptation interface is to be kept as simple as possible. On playback, it will be the task of either the decoder or the adaptation interface to reconstruct valid video or a valid video bit stream, including the correct timing of this video.

Even though at first the tape formatting for rudimentary trick modes seems trivial, some measures can be taken that significantly improve the fast playback picture quality for certain speed-up factors. In Section 3.4 the tape formatting for rudimentary trick mode playback will be discussed. The detailed discussion on the limitations and performance of rudimentary fast playback will be postponed to Chapter 4.

For rudimentary fast playback it is important to recall that in an MPEG video bit stream the smallest individually decodable entity is the *Slice*. Rudimentary fast playback pictures will thus be built up from recovered intra coded slices. For a 10 Mb/s MPEG video signal the slice length of long slices (45 macroblocks) will typically be around 30kbit. Obviously the usage of shorter intra slices (e.g. 15 macro blocks long) will be advantageous to the subject of rudimentary trick mode playback. With shorter slices on average a quicker synchronization on the slice headers will occur and as much data as possible will be recovered.

In Section 3.4 the discussion will be limited to the formatting of the bit streams onto the tape, with attention to recover whole slices, or larger groups of neighboring slices. The objective is to recover data sections with a burst length that is significantly larger than the expected slice length.

3.3.2 Dedicated Stream Fast Playback

The principle of helical scan recording only allows for a limited number of different recording bit rates. To record a specific service this means that a bit machine with a usable bit rate of at least the bit rate of the particular service must be used. Any bandwidth in excess of the service bit rate is free to be used for any other purpose, or otherwise it must be stuffed with dummy data.

The bit rate of a Standard Definition TV (SDTV) MPEG video service is expected to be in the range of 3.0 - 9.0 Mb/s. For such a service a bit machine operating at 12.5Mb/s (M4) will be used. The bit rate of some proposed High Definition TV (HDTV) MPEG video services is expected to be around 20 Mb/s [ATV94]. To record these signals a 25Mb/s (M2) recorder will be used. In both cases a spare bandwidth of about 20% exists.

The spare bandwidth can be exploited for dedicated fast playback signals. The principle is the following. The recording adaptation interface extracts a low bit rate dedicated fast playback bit stream from the normal play stream. The way in which this extraction is performed will be discussed in Chapter 5. One or more copies of the dedicated bit stream are put at strategically chosen positions on the tape. This must be done in such a way that, for a particular supported fast playback tape speed, the *entire* dedicated fast playback MPEG stream is extracted from the tape, i.e. irrespective of the scanner to tape phase the entire dedicated trick bit stream is recovered from tape. The difference from rudimentary fast playback is that the recovered signal is a priori defined and the resulting fast playback video signal has a pre-determined quality.

Two approaches can be identified for the tape formatting. The approach most commonly found in literature [Azad94, Boyc93, Lane93, Okam93, Yana93b] is the one where some degree of phase locking exists between the scanner and the tape during fast playback. The phase locking can be supported by a feedback control based on embedded signals or based on the actual data read. The alternative approach, which we have investigated, is one where the scanner and the tape are not locked in phase during fast playback. In Section 3.5 we will develop the tape format design methodology for dedicated stream fast playback.

3.4 Tape Formatting for Rudimentary Fast Playback

3.4.1 Plain Format

The adaptation of the MPEG video bit stream to a recorder has been discussed in Section 2.2.2. For normal operations, using a plain format, the functionality of the adaptation is quite straightforward. During recording the MPEG bit stream is passed to the recorder adaptation interface. At this interface the bit stream is packetized into sync-blocks and formatted for recording. At playback, the inverse adaptation interface re-generates a valid MPEG bit stream which is consequently passed to the decoder. Detected errors are signaled to the decoder using the dedicated start_codes.

During rudimentary fast playback, an invalid MPEG bit stream, consisting of bursts of valid information, will be recovered from the tape. At this point we assume that the valid data sections contain random sections of the bit stream and thus random sections of the video signal.

The nature of the bursts of data extracted can be deduced from the model developed in Section 3.2. Figure 3.11 gives the read mask for all possible phases. From Equation (3-2) it can be seen that the contiguous burst of data will never be longer than one track L_T=85kbit. The average fraction of data read from tape is given by Equation (3-1). For example, for the particular case where f_{READ}=0.33 for E=2, Q=0.5 and n=3.0, we can say that on average, the burst of valid data will be about 30kbit long. This burst length is comparable to expected slice length of long intra coded slices.

It is clear that with the short burst of data recovered, only very small sections of the picture are refreshed at one time. Ultimately, for very short bursts, there will be less than one slice per burst and the end of the slices will never be decoded. The resulting picture consists of very many small sections, each of which was updated at a different moment and each of which may have a different originating picture. Figure 3.12 gives an example of a fast playback picture that is recovered from very short data bursts. A tape formatting solution where longer contiguous sections from the bit stream are recovered would be desirable. In the next section such a tape format is developed.

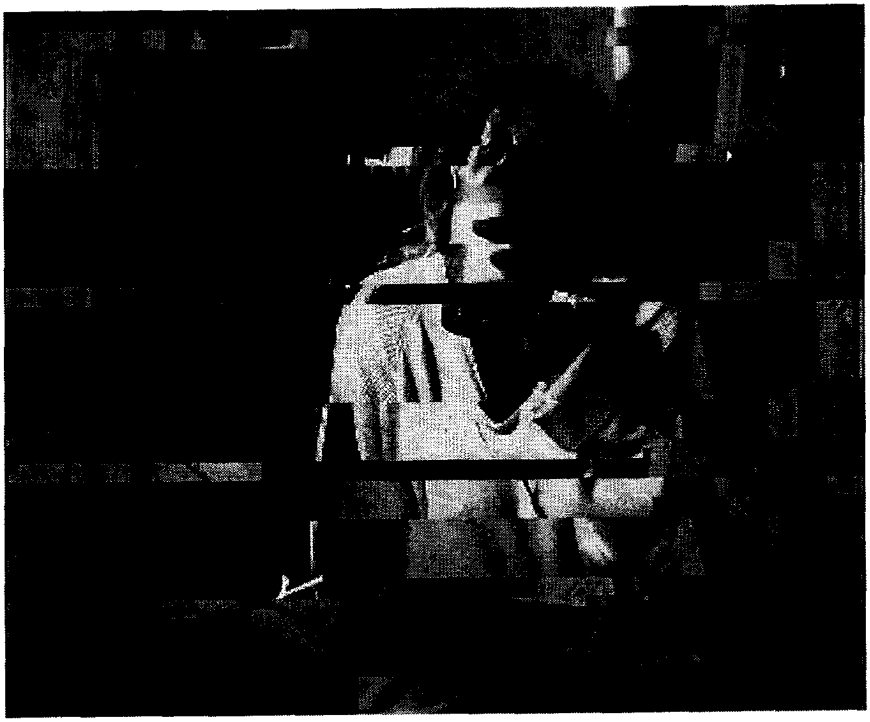


Figure 3.12: Example of rudimentary fast playback picture when very short data bursts are recovered from tape.

3.4.2 Contiguous Format

3.4.2.1 Adaptation Interface: Mapping

In order to be able to change the nature of the fast playback masks received from the tape, we will add some functionality to the adaptation layer. The main part of the adaptation layer becomes a mapping layer, which formats the bit stream for recording.

The mapping layer (Figure 3.13) contains a memory which allows us to shuffle the order of the packets (sync-blocks) during recording and to regenerate the correct order during playback. The size of the memory is equal to a Data Frame Structure (DFS) as this is the addressing range of the sync-blocks. A mapping table serves as a lookup table for sync-block numbers, where a sync-block is mapped to a specific tape position depending upon its number.

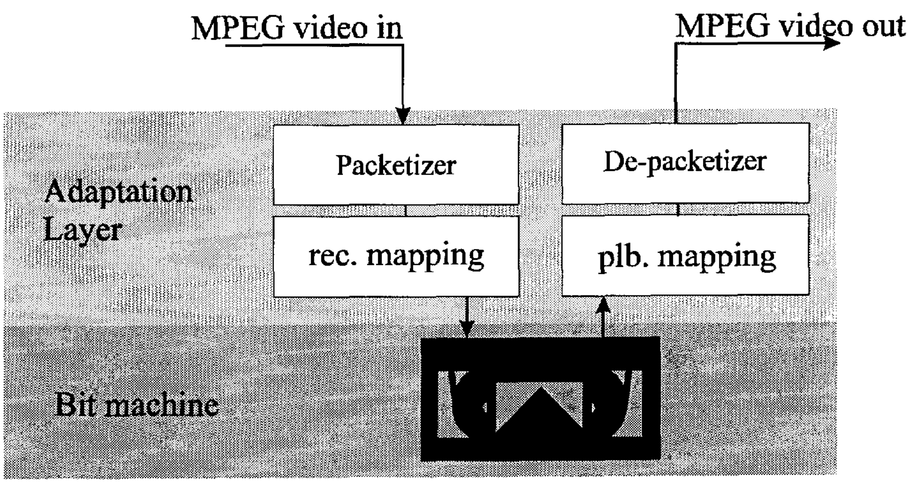


Figure 3.13: Adaptation interface.

Effectively, the mapping layer changes the read mask of the bit machine during fast playback and normal playback. The functionality of the mapping layer could therefore be considered to form part of the bit machine, yielding a machine with different characteristics during fast playback. In normal play, the mapping in is principle transparent, except for the effect of burst errors which are scattered over a DFS. Some extra error correction may therefore be required.

3.4.2.2 Transport stream mapping

Note that in the defined adaptation layer the sync-block numbers are used to re-map the sync-blocks. This is a significant advantage of using sync-blocks instead of transport stream packets which has not been discussed in Section 2.5.2. We will indicate what the added requirements will be when these sync-block numbers are not available.

Transport stream packets only have a 4 bit continuity counter; this is not sufficient to be used as a unique address in a DFS (e.g. 1656 sync blocks for a 12.5Mb/s machine). Therefore, when transport stream packets are recorded, a unique address like the sync-block number will need to be added to the recorded transport packets.

3.4.2.3 Contiguous Sections

From Figure 3.11 it can be seen that, for the particular trick speed of n=3.0, the sections read from the tape form a periodic pattern. It is straightforward to see that for the integer speedup factor n, the period of the pattern will be equal to n double tracks. The periodic structure of the bursts read from the tape can be exploited to design a format with longer contiguous bursts.

The periodic structure can be used to define the contiguous format which, for a particular speedup and irrespective of the phase, groups many individual data bursts within the DFS to a single burst. We aim at combining the periodic mask sections to a single contiguous burst in a DFS, where a burst can cross DFS boundaries. The length of the burst will depend upon the speed (n), the enhancement factor (E) and the Q-factor. Note that, as the mask is particular to a speed-up factor and to the considered bit machine type, the contiguous section re-mapping will be specific for a particular trick speed on a particular machine type.

We are free to define the formatting for any number of tracks with a minimum of n double tracks; the minimum is given by a single period of the read pattern periodicity. However, the bit machine gives us the possibility to address the sync-blocks uniquely for a fixed size DFS only. It is thus necessary to extend the formatting from the n double tracks to a full DFS.

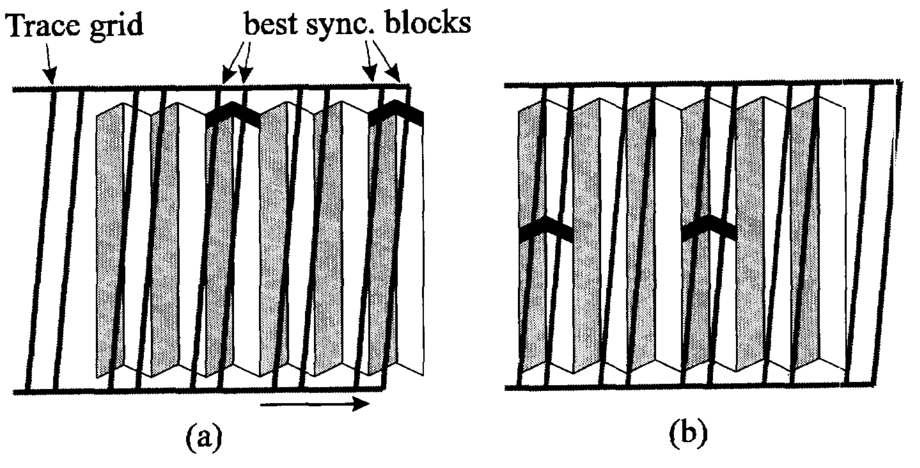


Figure 3.14: Geometric contiguous map design

The design of the contiguous map is based on the geometrical configuration as shown in Figure 3.14. The trace pattern, as predicted by the model for a particular speed-up and a particular bit machine, forms a trace grid. When this trace grid is

put on top of the tracks of a DFS we can identify the sync-blocks that are read with a maximal SNR, i.e the sync-blocks where the trace is exactly in the middle of the track. As an example, Figures 3.14 (a) and (b) show the trace grid (for n=3.0 on a M4 bit machine with E=2) put on top of a DFS in two positions.
The contiguous map design now is done as follows:

1. Choose a trace grid phase of p=0.0
2. Find the sync-blocks that are read with maximal SNR for each trace.
3. Map the identified sync-blocks one after another in the re-mapped DFS.
4. Increment the trace grid phase such that the next set of sync-blocks can be identified.
5. Continue from step 2 until all sync-blocks have been re-mapped.

It should be noted that this exercise is only necessary to obtain the mapping table; it is this table that will be used in the actual re-mapping of the DFSes in the interfaces.

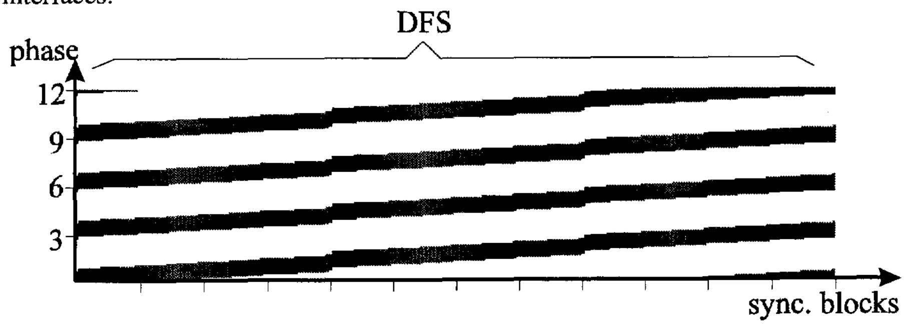


Figure 3.15: Re-mapped phase read diagram for a DFS (n=3.0, Q=0.5).

Using the obtained mapping table, Figure 3.15 shows the transformed DFS phase-read diagram of Figure 3.11. As can be seen from the figure, for n=3.0 and at a particular phase, the data which is read from a DFS forms a single contiguous burst. The burst in principle has a constant length, except for a few anomalies which are caused by the fact that not the entire track (full 180°) is used for actual data but some border sync-blocks are used for the second level error codes and other overhead information. As these blocks are not useful for our purpose they are not incorporated in the figure. This anomaly will have to be compensated for in a practical system by assuming that the Q-factor is lower than it actually is.

3.5 Tape Formatting for Dedicated Stream Fast Playback

In this section a tape format design method for dedicated stream fast playback will be developed. The main questions to be addressed with respect to the formatting of a dedicated fast playback stream on the tape are:

a) how many copies of the trick bit stream need to be recorded and
b) how are these to be formatted to the tape to assure that they are always entirely read during fast playback.

The first point determines the fast playback bit rate that can be supported by the device for a guaranteed signal at a certain speedup. This problem is closely linked to the specific mapping of the trick areas on tape that can be devised. In order to get a feeling for the possible solutions, four tape format design cases will be presented.

It should be noted that even though a solution is developed for some specific speedup factors, the techniques used are generic and can thus be applied to design the mapping for any desired speedup factor. In a practical situation employing dedicated fast playback streams inherently limits the fast playback to fixed speed-up factors (forward or reverse).

3.5.1 Two Layered Adaptation

Two different sub-layers of the service to recorder adaptation can be identified (Figure 3.16). In the outer layer the recording adaptation interface extracts one or more appropriate fast playback bit streams from the incoming bit stream. The outer playback adaptation interface simply needs to select the appropriate bit stream to be transmitted to the decoder.

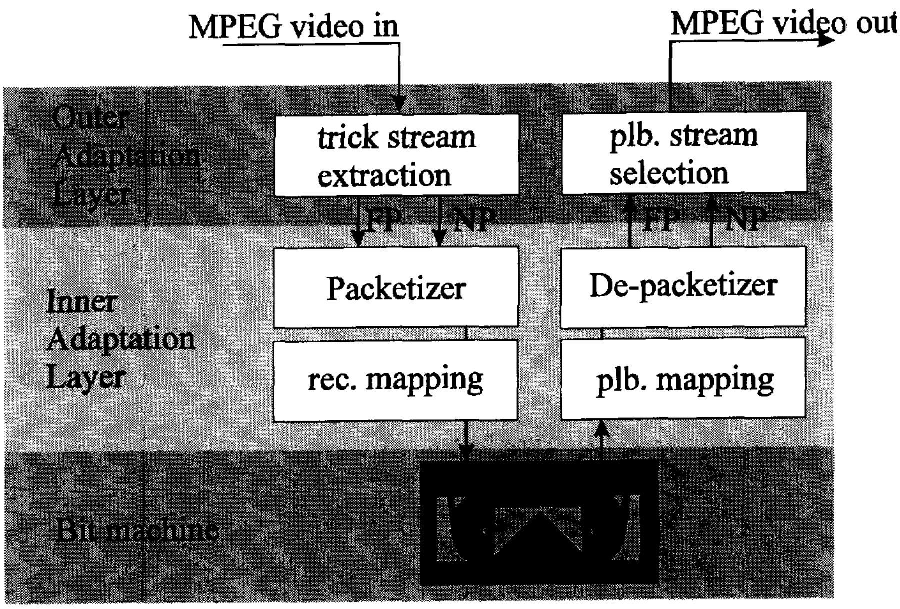


Figure 3.16: Two layered adaptation interface.

The packetizing and formatting of the bit streams onto tape is performed in the inner (mapping) layer. The functionality of this adaptation layer is similar to the continuous mapping layer of the rudimentary fast playback. The playback mapping layer will invert this formatting to generate one of the original bit streams, the normal stream (NP) in case of normal play and (one of) the dedicated fast playback stream(s) (FP) in case of fast playback.

3.5.2 Mapping Design for System without Phase Lock

3.5.2.1 Base Map: Contiguous

In order to simplify the design of an appropriate fast playback formatting, a first level mapping is defined which is identical to the contiguous mapping of the rudimentary fast playback. The contiguous mapping is done by the use of a mapping table, transforming the addresses of the sync-blocks between the inner adaptation layer and the recorder.
The advantage of this initial contiguous re-mapping is that the external pattern of the bursts of data for a DFS is less dependent on the specific trick speed; there will always be a single burst of data within the DFS. Because the length of the burst will depend upon the speed (n), the enhancement factor (E) and the Q-factor, the actual trick stream format design can be parameterized such that the interfacing approach and the format design is independent from the speedup factor.

For the second level mapping it is important to note that a single static mapping table must be used for every DFS and therefore, the formatting will be identical for every DFS. This will only be possible if, for every consecutive DFS, both the position and the size of the recovered sync-block bursts (S_{BURST}) is constant. This implies that the scanner to DFS phase at the beginning of a each consecutive DFS is required to be constant.
Only a limited number of speed-up factors, which are determined by the DFS size, will satisfy the constant S_{BURST} requirement and can therefore be considered. For the DART with the M4 bit machine, the DFS size is 12 tracks: only n=3, n=4 and n=6 can be supported as fast forward speeds and n=-3, n=-4, n=-6 can be considered as fast reverse speeds. For the DART with the M2 bit machine, the DFS size is 24 tracks, such that n=12 and n=-12 can be added to the list.
Note that the size of the DFS for the different machines is historically defined and is not governed by the desired speedup factors. Our adherence to the DFS size and the DFS based interfacing is only governed by the need for experimental verification. The formatting techniques developed are, however, equally applicable when the DFS constraint is loosened to allow for more different speedup factors.

3.5.2.2 Design for Single Fast Playback Stream

The first goal in the format design is to multiplex a single fast playback stream into the DFS such that it is guaranteed to be read, irrespective of the scanner to tape phase (p).

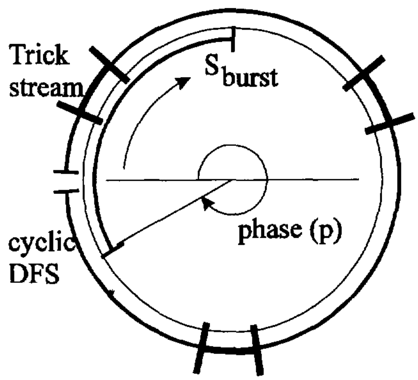


Figure 3.17: Cyclic DFS burst.

Because for each DFS the same burst of sync-blocks will be read, the read mask is periodic with a period equal to a DFS and we can symbolically draw a cyclic DFS as shown in Figure 3.17; the DFS is defined to be the full circle. The contiguous burst of data (S_{BURST}), read from the DFS, forms a section of this circle at a position determined by the scanner to tape phase (p).
Given that the burst length (S_{BURST}) is constant, the size of the data burst is determined by the average fraction read of Equation (3-1). The burst length of Figure 3.17 thus equals:

$$S_{BURST} = E \cdot \frac{Q}{|n|} \cdot S_{DFS} \qquad \begin{array}{l} E=1,2 \\ n=\pm 2,\pm 3,\pm 4,\pm 6 \\ 0 \le Q \le 1 \end{array} \qquad (3\text{-}3)$$

where S_{DFS} is the size of the DFS expressed in sync-blocks. We can conclude that the multiple copies of the same data must start no further than S_{BURST} from each other. Consequently, the number of required copies (m) of the same fast playback data is determined by the number of times S_{BURST} fits into S_{DFS} as follows:

$$m(n,Q) = \left\lceil \frac{|n|}{E \cdot Q} \right\rceil \qquad (3\text{-}4)$$

where the division is rounded upwards to the nearest integer. It does not matter where exactly the trick streams are placed, as long as the beginning of the m copies are separated no more than S_{BURST} sync-blocks from each other.
Figure 3.18 shows the dependence of m on the Q-factor for three different speeds (n=±3,±4,±6). As an example we note that for the specific case of n=3.0 and Q=0.5, the number of required copies of the same fast playback bit stream portion is m=3. It is observed that Q=0.5 is a border case and for reasons of robustness m=4 may be preferred.

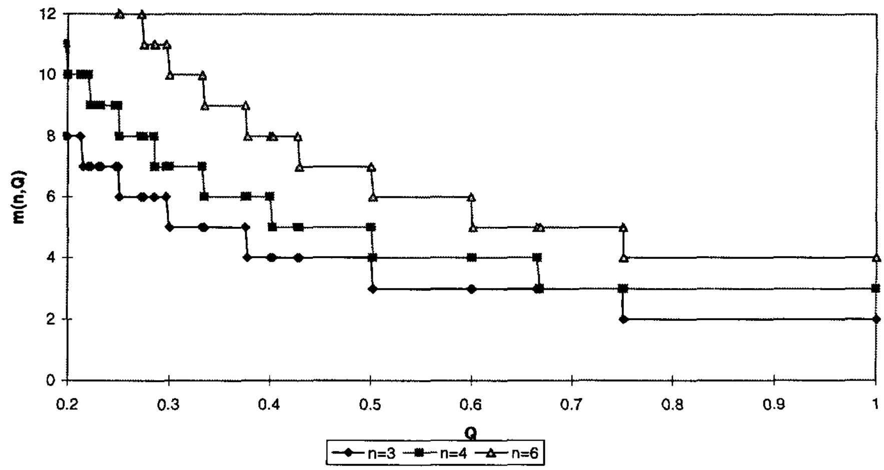


Figure 3.18: Amount of required copies m(n,Q) as function of Q-factor and n=±3,±4,±6

Consider what happens if a different speedup factor is used such that the size of S_{BURST} is not equal to the average for every DFS. It now becomes impossible to design a fast playback formatting based on the analysis of the average amount of data read from the tape. The only remaining possibility is to perform a worst case analysis which will lead to a significant increase in the number of required fast playback copies.

Now that the required number of copies m(n,Q) are known we can calculate what the available bit rate will be, given a certain spare tape bandwidth to be used by the dedicated fast forward bit stream (R_{SPARE}). The available bit rate for the fast playback bit stream (R_{TRICK}) will be:

$$R_{TRICK} = n \cdot \frac{R_{SPARE}}{m} \qquad (3\text{-}5)$$

For a speedup of n=3 with m=3, R_{TRICK} is equal to the spare bandwidth R_{SPARE}. In a practical system, this means that when recording no more that 10Mb/s video on a 12.5Mb/s (M4) machine, a 2.5Mb/s dedicated trick bit stream can be recovered in a system with unlocked phase.
Figure 3.19 shows the resulting phase-read diagram for a DFS when the appropriate mapping table is concatenated with the contiguous mapping table and the resulting map is applied to create the fast playback areas. It can be seen from this figure that, irrespective of the phase, for every DFS a known data section is guaranteed to be read.

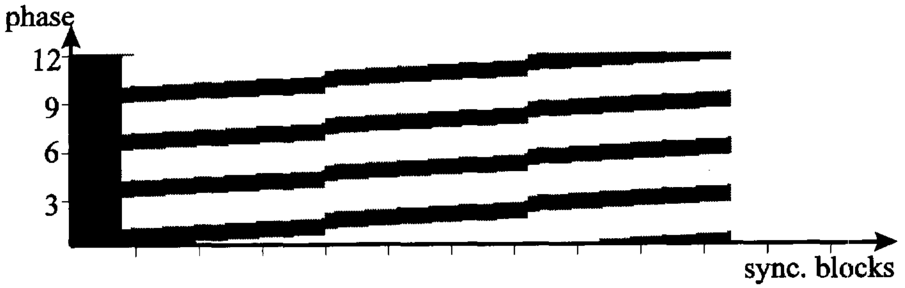


Figure 3.19: Phase-read diagram for fast playback with a single fast playback stream

Once the fast playback areas have been identified, the contiguous re-mapping needs only to be applied to the specific fast playback areas. The normal play stream can be written to the normal play section of the DFS using the plain format. In Figure 3.20 the normal play sections read during fast playback are not contiguously re-mapped. The advantage of not re-mapping the normal play section is that less memory will be required in the mapping interface because only the fast playback data will be buffered for the entire DFS period.

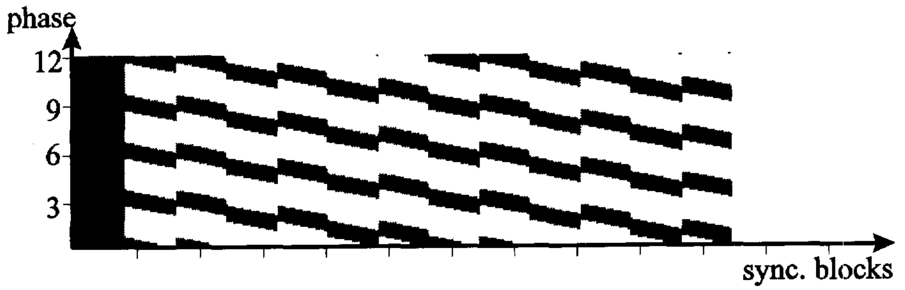


Figure 3.20: Contiguous mapping for trick section only for a single fast playback stream map

3.5.2.3 Design for Two Fast Playback Streams

Obviously the method described above can also be used when multiplexing more than one fast playback bit stream into the DFS. In that case, the spare bandwidth needs to be divided over the multiple bit streams. As an example, we have performed the formatting for two fast playback streams, one for $n=n_3=3$ and the other for $n=n_6=6$. The spare bandwidth of 20% is divided into two parts of 10% (i.e. 1.25Mb/s) each. For both cases, m is equal to n, such that the resulting two fast playback bit rates are 1.25Mb/s.

Figure 3.21 and Figure 3. 22 show the resulting phase-read diagrams when a single mapping table is designed to multiplex the two trick streams into the DFS. Note that as two different contiguous re-mappings need to be performed, they should operate on the appropriate fast playback data sections only.

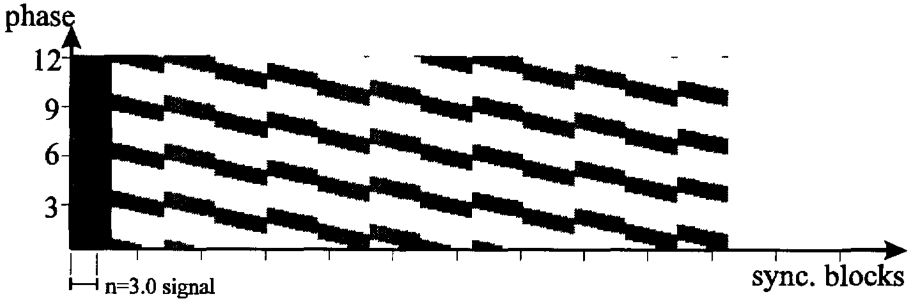


Figure 3.21: Phase-read diagram for the map that supports two fast playback streams (n=3.0, Q=0.5)

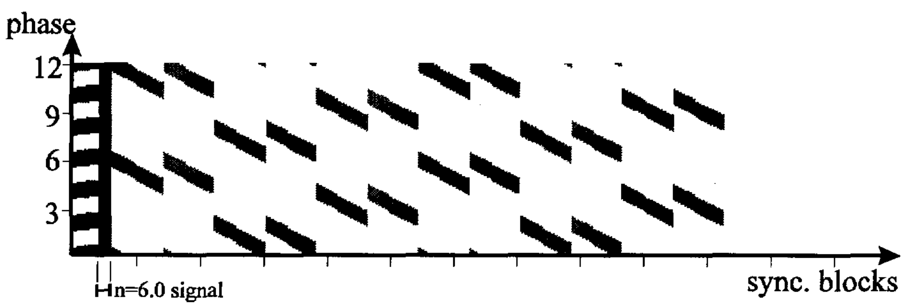


Figure 3.22: Phase-read diagram for the map that supports two fast playback streams (n=6.0, Q=0.5)

We can see that if, before re-mapping, the n_3 stream is formatted in the lower (10/3)% section (10% of the capacity, requiring m=3 copies) of the DFS and the n_6 is formatted in the (10/6)% section (10% of the capacity, requiring m=6 copies) above that, both sections are guaranteed to be read at the appropriate speeds.

3.5.2.4 Design for Two Speed-up Factors with Single Fast Playback Stream

A special mapping table for n=3 can be designed with which it is possible to recover a single fast playback signal at different playback speeds: n=3, n=-3 and n=6. The contiguous mapping forms a single burst for n=3 and n=-3 and two separate bursts for n=6. The entire spare capacity of 20% is thus available for one trick bit stream. The copy-factor (m) is determined by the worst case (n=6.0) such that m=6. For the given system, where the spare bandwidth R_{SPARE}=2.5Mb/s, the bit rates of the trick streams are $R_{TRICK,6}$=2.5Mb/s and $R_{TRICK,3}$=1.25Mb/s for n_6 and n_3 respectively. Figure 3.23 and Figure 3.24 show the resulting phase-read diagram when the trick stream is read out at n=3 and n=6 respectively.

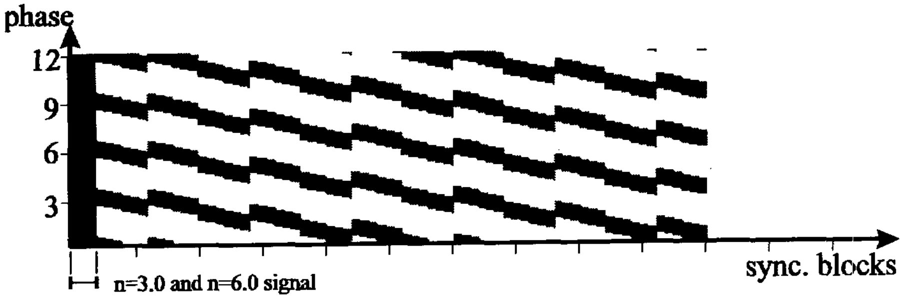


Figure 3.23: Phase read diagram for map that supports two speedup factors with a single stream (n=3.0, Q=0.5)

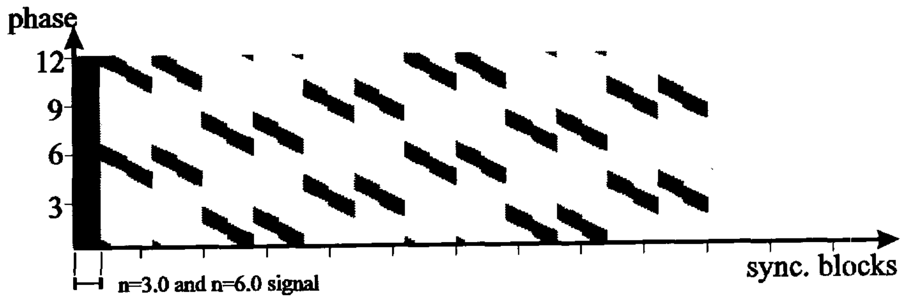


Figure 3.24: Phase read diagram for map that supports two speedup factors with a single stream (n=6.0, Q=0.5)

Both fast playback signals will have the same content. However, the bit rate and the frame rate of the n_6 fast playback bit stream will be twice that of the n_3 signal. When compared to the previous solution there is no difference for the n_3 signal. As the same bit stream will have a different frame rate and bit rate on a different occasion, the playback adaptation interface will have to assure that the relevant fields in the sequence header are filled appropriately. Furthermore, for the reverse playback at n=-3 some re-shuffling at a higher adaptation level, collecting the entire frames, will be required for a standard decoder to be able to decode the reverse stream. This matter will be elaborated in Section 3.5.3.

3.5.2.5 Hierarchical Medium

Observing the phase-read diagrams for the design for two fast playback streams, it can be seen that the n=6.0 trick stream can be read at the n=3.0 trick speed. For specific well-chosen formattings this may be the case. This formatting creates a scalable medium, where at different tape speeds, different layers of the hierarchy are extracted. The formatting used for Figure 3.21 and Figure 3.22 can be interpreted as a three level hierarchy: a n_6 level, a n_3 level and the normal level. At

n=6 only the n_6 level is read, at n=3 both n_6 and n_3 levels are read and at n=1 all three levels are read.
A matching scalable MPEG-2 stream could exploit this medium hierarchy, using both temporal and spatial/SNR scalability. Temporal scalability could be useful to provide a constant frame rate at the different speeds. Alternatively, a different frame rate could be used for the different playback speeds as was done for the design for two speed-up factors with a single fast playback stream. Spatial or SNR scalability will be the main tools to control the individual bit rates of the different layers.
As it cannot be expected that an available scalable broadcasted stream will fulfill the strict requirements on the different scale levels imposed by the recorder, the scalable solution may not be useful for transparent recording. It may, however, be attractive for those cases where recorder specific encoding is performed and the decoder is capable of handling the scalable syntax.

3.5.3 Fast Reverse Stream

From the format design discussion we can conclude that the formatting of a guaranteed fast reverse stream is almost identical to the formatting of a forward stream. However, where the dedicated forward stream can simply be divided into DFS packet groups which are formatted into the DFS structures, some extra bit stream transformation will be required before recording the reverse stream. The main problem with the reverse stream is that, where it is created in the forward direction, it must be played back in the reverse direction.

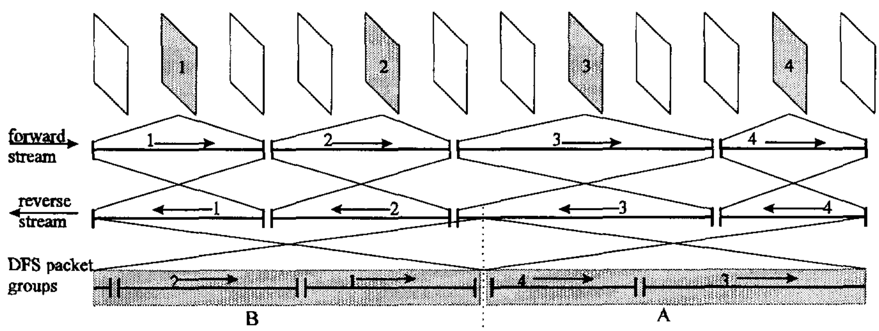


Figure 3.25: Fast reverse stream transformation (n=3.0, T_m=1)

Figure 3.25 symbolically shows the typical required transformation steps for the fast reverse stream. The individually extracted pictures form the presentation units. When these pictures are concatenated the fast forward stream is formed. The presentation units of the reverse stream are exactly the same as those of the forward stream for the same speedup factor and frame rate. For the reverse stream the individual pictures must however be transposed, such that in reverse order the first packet of each picture is encountered first.
For a helical scan recorder, the order in which data is read from a track is identical for both forward and reverse playback. In our architecture with a mapping interface,

this effect is even stronger and during reverse playback the data from the individual DFS's is read in the same order as for forward playback. Therefore one more transposition of the reverse stream must be performed. The reverse stream is divided into DFS packet groups, where each packet group will be formatted into the DFS according to the methods of Section 3.5.2. Within each DFS the packets must be transposed, starting to record the last packet and ending with the first packet. This second transposition can easily form part of the mapping performed by the adaptation interface.

On playback DFS A is read first (Figure 3.25), starting with picture 4 and most of picture 3, after which DFS B is read and the remaining of picture 3 is read followed by pictures 2 and 1. The performed transformation thus assures that on playback the adaptation interface needs to make no distinction between forward playback or reverse playback and a unique operation can be defined.

As a final note we would like to point out that for the "Design for Two Speed-up Factors with Single Fast Playback Stream" approach (Section 0) no specific reverse stream is recorded and the necessary transformation can not be performed. This will require the playback interface to reconstruct a valid bit stream by re-ordering the packets on reverse playback. This task could be assisted by aligning the dedicated frame boundaries with the DFS boundaries.

3.5.4 Tape Formatting for System with Phase Lock

For a more advanced tape system, where the phase of the scanner is locked to the tape during fast playback, the techniques will not be too different from those discussed above. However, in most cases fast playback bandwidth advantage can be gained from the fact that the phase locking is possible. We distinguish two types of phase locking.

The first is a system where the scanner to tape phase is locked to the beginning of the tracks on the tape. In such a system there are still different possibilities with respect to the phase within a DFS on tape; all the even integer phases (p=0, 2, 4) of Figure 3.15 still apply. Limiting the possible phases will in general reduce the number of required copies (m) of the same data on the tape. In particular for the higher speedup factors, the required overhead will be reduced. As an example, consider the following. Analyzing the periodic phase-read pattern for both $n_3=3.0$ and $n_6=6.0$, it is easy to deduce that for both these cases m=3. For n_3 there is no difference from the lock free system, for n_6 the available trick bandwidth will be doubled.

The second type of phase locking is that where the scanner phase is locked to the beginning of a DFS on the tape. In such a system ultimately m=1 for all speedup factors. In practice some more copies will be required to accommodate smooth transition from one speed to the other.

Systems that support phase locking yield a larger available fast playback bandwidth such that far better fast playback picture quality can be achieved. In practice it will often be more attractive to use this spare capacity to support more different speed-up rates, including a number of dedicated reverse rates.

4. Rudimentary fast playback

4.1 Introduction

In the previous chapter on tape formatting for fast playback, rudimentary fast playback was stipulated to be a feasible approach. The basic idea of rudimentary fast playback is that a fast playback sequence is formed based on the shreds of recovered valid information. The elegance of this solution is the low hardware complexity, in particular at the recording side. However, the extraction of a valid video signal from the invalid bit stream recovered during fast playback is complicated because:

- Only fragments of the normal play stream are recovered during fast playback; these fragments must be collected and packaged to form a valid MPEG stream.
- The video stream is encoded using picture types (P and B) that require other, previously transmitted pictures to be decoded. We have no guarantee that these referenced pictures, or even more so the referenced picture sections, are read from tape and decoded. Consequently, only the intra (I) coded pictures are usable.
- Intra-coded slices are not individually recognizable, i.e. in general it is necessary to decode the picture header to be able to interpret the slices of that picture.
- Some un-recovered syntax layers may contain information necessary to decode recovered lower layers.

The fast playback video has to be based on intra encoded *Slices* that are recovered. As the *Picture* headers are in general not recovered, it is important to be able to recognize those slices that are intra encoded. The MPEG-2 syntax contains a provision in which an optional flag of the slice header can indicate that a slice is intra coded: the Intra-slice flag. For the rudimentary fast playback to yield any results at all, the Intra-slice flag should be set by the encoder or alternatively, the recording adaptation interface should set this flag. Note that the second option requires the adaptation interface to recognize the appropriate parts of the bit stream, which yields a significant increased complexity.

As no presentation units (pictures) exist anymore, new ones have to be formed by grouping the recovered slices in new pictures. In the rudimentary fast playback signal each decoded picture is build up from slices that belong to different pictures of the normal stream. The quality, measured in SNR per slice, of the fast playback video, is identical to that of the normal play video. In this respect, rudimentary fast playback is a good solution. However, not every part of the picture will be refreshed at the same time and there is no control over which slice is recovered. It is possible that the same slices are update all the time and some slices are hardly updated at

all. Rudimentary fast playback therefore strongly relies on an even distribution of the updates over the entire picture such that a reasonable refresh time can be expected for all slices. The main free parameters in the refresh distribution are the speedup-factor, the tape formatting and the GOP structure used.
The resulting quality will be evaluated by simulation in this chapter and it will experimentally be evaluated with the hardware verification model in Chapter 6. In the evaluation the visual picture quality, two requirements on the refresh mechanism can be identified. First of all the updates of the slices must be equally distributed over the whole picture, i.e. all macroblocks must be updated at an appropriate rate. This requirement is the most important in this rudimentary fast playback analysis.
Secondly, neighboring macroblocks should be updated as much as possible at the same time, i.e. the visual quality of the fast playback picture will benefit from the fact that large areas originate from the same normal stream picture. This second requirement has been the motivation for the introduction of the contiguous format in Chapter 3. In this chapter we will evaluate the effect of this tape format.
We will define an objective measure to verify each of these two requirements for different formatting and system solutions.

In Section 4.2 some of the limitations of the rudimentary fast playback approach will be discussed. In Section 4.3 we will analyze the performance using a model of the recorded bit stream. In Section 4.4 the results for an actual video signal will be evaluated by simulation. Finally in Section 4.5 the algorithm for rudimentary fast playback will be translated to a system architecture of a playback adaptation interface that generates a fast playback video stream which is usable by any MPEG video decoder, irrespective of its fast playback provisions.
All discussions will be based on the 12.5Mb/s (M4) machine, with a DFS of 12 tracks. The recorded normal play signal will have a bit rate below 10Mb/s such that some dummy packets will need to be inserted to match the two bit rates.

4.2 System Limitations

The Intra slice based rudimentary fast playback solution is a general procedure that will work for many MPEG encoder (normal stream) implementations. However, when certain options of the syntax toolset are used then problems can be expected. Some of these problems may even occur within the *Main* profile, *Main* level constrained implementation. We will briefly discuss the most relevant problems that may occur such that the constraints on the standard's toolset usage can be identified.

4.2.1 Field Pictures

When field pictures are used then one picture represents a single field, not a frame. The layered structure of the field pictures is identical to that of a frame picture; the only real difference is that the picture rate is doubled, updating a field at a time.

When slices of a field picture are updated in rudimentary fast playback, then a stripe of 16 lines from one field is updated; the co-located 16 lines of the other field will remain unchanged. In fast playback it is possible that a certain slice is not updated for a longer period of time. Two co-located slices can thus originate in different frames and possibly in different scenes. This interleaving of sections from two different scenes will visually be highly disturbing. Even if both fields were updated in the same scene but at a different time, the motion in the video will in general degrade the fast playback picture.
The solution to the problem for field pictures is to display one field only during the fast playback. All the recovered slices should thus be considered to belong to the same field, e.g. the odd field, and this field should be displayed twice per frame period. An interesting point is that the transport stream supports the signaling of this display requirement by the inclusion of a *field_id* flag in the fast playback byte of the PES packet header. This mechanism can be used to signal the decoder what operation is requested.

4.2.2 Changing Higher Layer Parameters

In the principle of rudimentary fast playback only slice information will be interpreted. However, parameters of a higher level may be of importance to this interpretation. When these parameters change during the course of fast playback then they could, and often will, be missed by the adaptation interface, which will have the following consequences.

4.2.2.1 Frame/Field Picture Toggle

The higher layer information is important when the encoder uses both field and frame pictures and toggles between these within a single sequence. When such a toggle occurs during fast playback then the slices will be misinterpreted; frame slices could be interpreted as fields and vice versa. The fast playback video will be build up of both types of slices such that the resulting picture will not be consistent.
If the required higher layer information is not guaranteed to be read then the only remaining solution is to require from the encoder that only one picture structure is used, either field or frame pictures.

4.2.2.2 Alternative Weighting Matrix

An important example of missed higher layer parameters may occur with the transmission of an alternative quantizer weighting matrix in the picture header. If such a weighting matrix is missed, the weighted de-quantization will be done differently than intended.
The usage of alternative weighting matrixes will largely depend upon the particular encoder implementation. Simple encoders will never change this matrix and stick to the default. Alternatively, high end encoders could adapt this matrix for every scene or even every picture. The extent in which this matrix will change is also implementation dependent but, as the matrix is averaged for at least a whole picture, we expect the variation to be relatively low.

Some experiments were performed to evaluate the effect of decoding with an incorrect quantization matrix. The different quantization matrices mainly affect the way the higher frequencies are quantized. An incorrect matrix has the effect of either incorrectly enhancing or surpressing the high frequencies. The picture itself will, however, always remain recognizable. For the purpose of fast playback we conclude that visual search remains possible, even with changing quantization matrices.

4.2.2.3 Alternate Q, Alternate VLC, Alternate Scan

A comparable problem of changing picture layer parameters is caused when the encoder uses some of the alternate tools: alternate quantizer table, alternate coefficient VLC table and alternate scanning (field zigzag). If any one of these alternate options is used and the appropriate flag is not recovered during fast playback then the bit stream will not be decodable at all.

It should be noted that this kind of alternate tools may be selected by the encoder at any picture but it is expected that these options will either be used for an entire sequence or not at all. If the option is already selected before the fast playback is effected then there will be no problem in using this tool. We therefore have to require that for the whole sequence the same scanning, quantizer table and VLC tables are used.

4.2.3 Hybrid Solution

If the requirements on the constant decoding parameters are too limiting then a mechanism must be provided which guarantees the reading of the higher layer header information. This mechanism could be seen as a combination of the rudimentary fast playback and the dedicated fast playback stream approach. The combined (hybrid) approach consists of a dedicated fast playback stream extracted from the original bit stream that contains all MPEG-video layers down to the Intra slice layer, but excluding this layer. This low bit rate dedicated stream can be formatted for guaranteed reading through the DFS using the techniques of the dedicated fast playback stream formatting but requiring far less overhead sync-blocks. The fast playback can now work on the principle of rudimentary fast playback decoding, but using the information from the dedicated stream to monitor changing global parameters. In the dedicated stream extraction, some stuffing provisions will have to be taken to maintain the synchronization between the dedicated stream and the normal play stream.

The advantage of the hybrid solution over rudimentary fast playback is the flexibility for adapting *Sequence* and *Picture* parameters. The disadvantage is the increased complexity of the adaptation interface.

When compared to the dedicated fast playback stream formatting the advantage of the hybrid solution is the decreased overhead bit rate such that far more fast playback speeds can be supported. The disadvantage is that the fast playback picture quality is not pre-determined and depends on statistical parameters.

In this chapter we will assume that the *Sequence* and *Picture* parameters will not change during the recording. In that case the headers will not be required.

4.3 Macroblock Updates for Different Speed-up Factors

4.3.1 Macroblock Age

In Figure 4.1 the principle of the rudimentary fast playback is illustrated for three consecutive fast playback pictures. For each fast playback picture the updated slices are shaded; the slices that are not updated are simply copied from the previous picture. A fast playback picture consists of both new and older slices. It should be noted that usually the last slice of an updated section will not be completely updated; the last macroblocks of the slice will probably be missing.

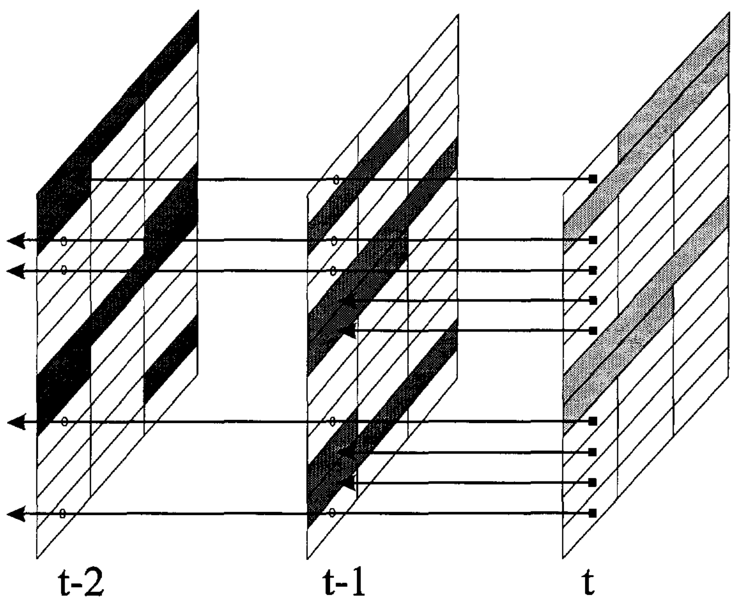


Figure 4.1: Macroblock age

It is desirable that on average all macroblocks have an equal probability of being updated such that all macroblocks are regularly updated. We requires a measure for the evaluation of the distribution of the macroblock updates for different mappings and system configurations. The *macroblock age,* which can be determined for every macroblock of each fast playback picture at time t, is an appropriate measure to analyze the distribution of the updates. The macroblock age is defined as the number of fast playback frame periods the particular macroblock was not updated.
We propose to use the average macroblock age (called *picture age*), to identify if all macroblocks are regularly updated. If one or more macroblocks are never updated then their age is infinite and the resulting picture age will be infinite. In our

practical measurement the age of non-updated macroblocks will however be limited to the amount of pictures that precede the current picture and the picture are will simply have a large value. If the refreshes are equally distributed over the entire picture and all macroblocks are updated regularly then the picture age will mainly be governed by the availability of Intra slices in the bit stream.
Figure 4.2 shows the simulated average picture age as a function of the speedup factor n, for two sequences with two different GOP structures, where the average is taken over a 200 frame sequence. The way in which this simulation was carried out will be discussed shortly. For now, the important observation is that, for a particular GOP structure, the picture age shows little variation for most speedup factors except for several specific speeds, where very large picture ages are measured. This observation suggests that, using the basic rudimentary trick mode scheme, for certain speeds some macroblocks will only scarcely be updated.

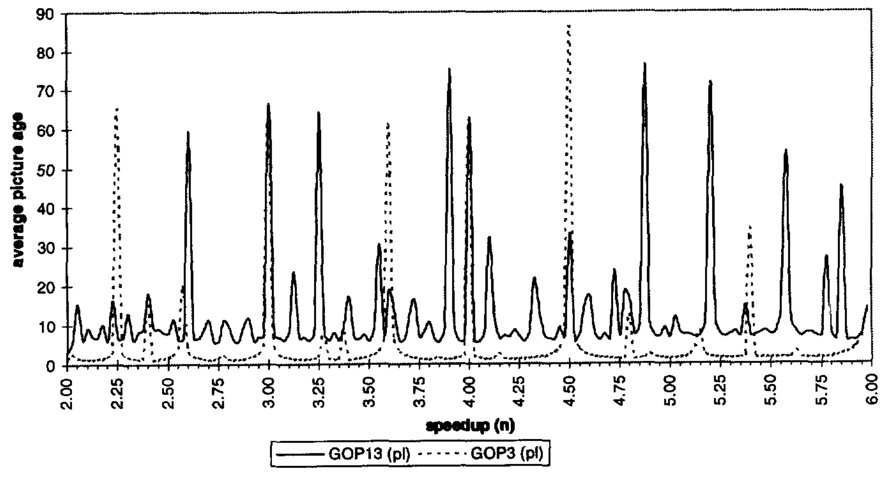


Figure 4.2: Average picture age as function of speedup for two different GOP lengths

4.3.2 Non-Random Update Mechanism

We have previously pointed out that one of the problems with fast playback on a compressed signal is that there is no direct relation between the position of bits on the tape and the specific picture section these bits encode. In the rudimentary fast playback approach we initially assume the *random* recovery of intra coded slices. Under this assumption a statistical distribution of the updated slices would seem to exist. In practice, however, this will often not be true. Depending upon the particular implementation of the rate control of the encoder, some correlation between the tape position and the video content will exist. In this section we analyze what the effect of this correlation is on decoded rudimentary fast playback streams. In particular the effect on the macroblock age distribution for the fast playback pictures will be evaluated; we expect to explain the reason of the peaks in Figure 4.2.

The joint effect of two kinds of periodicities cause the non-random update mechanism. The periodicity in the GOP structure causes Intra slices to be recorded at more or less regular intervals on tape. The read mask periodicity causes periodic sections to be recovered from tape.

4.3.2.1 GOP Periodicity: Model for Bit Stream

The MPEG standard does not dictate the number of pictures (L) in a GOP; in practice the GOP sizes can range from one picture per GOP to about 15 pictures per GOP. Many hardware encoders use a fixed group of picture (GOP) structure, although in high end (often software based) encoders the GOP boundaries will be aligned with the scene boundaries. Typical examples of the GOP lengths are GOP3, a 3 picture GOP with the I-B-P structure and GOP13, a 13 picture GOP with the I-BBP-BBP-BBP-BBP structure.

In order to analyze the effect of the GOP length we consider a fixed GOP picture length (L) with a rate control tuning to a fixed number of bits per GOP. In our analysis of the GOP periodicity the track rate f_{TRACK}=(12.5*12)=150 Hz (12 tracks/DFS, 12.5Hz DFS rate), the frame rate f_N=25Hz and the recording bit rate R=12.5Mb/s=L_T*f_{TRACK} are the basic parameters, where L_T is the amount of bits per track. The average number of bits per GOP period (T_I) equals:

$$T_I = \frac{R \cdot L}{f_N} = \frac{f_{TRACK}}{f_N} \cdot L \cdot L_T = 6 \cdot L \cdot L_T \text{ (bits/GOP)} \tag{4-1}$$

The interpretation of this period T_I is as follows: when we observe an intra slice in the bit stream at a position x then it is likely to find the same intra slice in a picture at a position x+T_I in the bit stream.

In order to evaluate the effect of the periodicity, as a function of the speedup n and the GOP length L, we define a model of the bit stream. In the bit stream model, Equation (4-1) is not interpreted as the average of the GOP periodicity. Instead it is the deterministic period of a GOP of length (L). The model bit stream will thus contain GOPs of a fixed length.

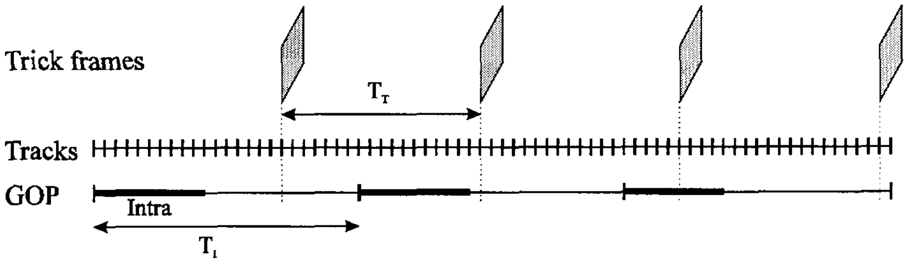


Figure 4.3: Model of bit stream

Furthermore, the model is extended with a deterministic intra slice position. Each GOP of the model starts with an Intra picture. An Intra picture of large GOPs uses 12 tracks, for shorter GOPs an Intra picture makes up half of the GOP data. Within the Intra picture the macroblocks use an equal and constant amount of bits.
Figure 4.3 illustrates the bit stream model. Fast playback pictures are formed with the period of T_T. All periods are expressed in terms of bits on tape instead of time, as the time base will be different in normal play and fast playback.
The importance of this deterministic bit stream model is that the worst case for the periodicity can be studied. An exact periodicity (period T_I) exists in position of the intra coded macroblocks in the bit stream. We are thus able to numerically predict what image sections will be recovered from the intra coded image as a function of the speedup (n), given a certain tape formatting and a fixed GOP length (L).

For many constant bit rate MPEG video encoders the rate control will be such that the buffer fullness at the beginning of the GOP's will be more or less constant, provided that the scene characteristics do not change drastically. Obviously this is dependent upon the particular implementation of the rate control, but for many rate controls it is a design constraint [With93, Ramc93, Fri93]. An encoder with a very tight rate control will approach the model quite closely. In literature encoding mechanisms can be found that approach the objective of a fixed number of bits per GOP with an error less than 0.1% [Kees94]. For instance, for a 10Mb/s video stream with the GOP length L=13, the error will be less than 5kb.
The premise that each macroblock used the same amount of bits does not hold for practical systems. However, in many slowly changing video scenes there will be a strong correlation between the amount of bits used for a certain macroblock position in the consecutive Intra pictures. As such, the period T_I from the slices in one GOP to the next will still be relevant.

4.3.2.2 Read Mask Periodicity

Assuming the plain formatting, the fast playback read mask from the tape and from the bit stream is periodic. For example, as discussed in Chapter 3, for a speed-up which is an integer fraction of the DFS track size (n=±3.0, n=±4.0, n=±6.0), the same section is read from every DFS and thus the period of the read mask (T_R) is equal to a DFS size of 12 tracks. In more general terms for a speedup factors (n) a periodicity will exist with a period

$$T_R = n\cdot(2\cdot L_T) \quad \text{(bits)} \tag{4-2}$$

However, the mask read from the tape will only be identical for consecutive periods if n is an integer, i.e. the phase of the heads to the track is identical for every period.
The read mask determines what sections of the data stream are recovered. A further matter of importance is how these sections are grouped in fast playback pictures to form a fast playback video sequence. We assume the fast playback frame rate to be 25Hz, i.e. the frame rate of the fast playback signal (f_T) is equal to that of the normal play signal (f_N). Given the average of R/f_N bit per normal play picture, the

number of bits on tape per fast playback picture is dependent upon the tape speed-up factor n as follows:

$$T_T = \frac{R}{f_N} \cdot n = \frac{f_{TRACK}}{f_N} \cdot n \cdot L_T = 6 \cdot n \cdot L_T \text{ (bits/picture).} \qquad (4\text{-}3)$$

4.3.3 Prediction of Macroblock Update

Using the bit stream model we can now predict how consecutive fast playback pictures will be updated as a function of the speedup factor. Figure 4.4 shows the fast playback picture refreshes for two specific speeds with a GOP13 normal playback stream. In the figures, an entire picture is symbolically plotted on a vertical line, where the top of the line signifies the first macroblock and the bottom of the line signifies the last macroblock. Horizontally consecutive pictures are plotted. For each fast playback picture, the refreshed macroblocks are marked.

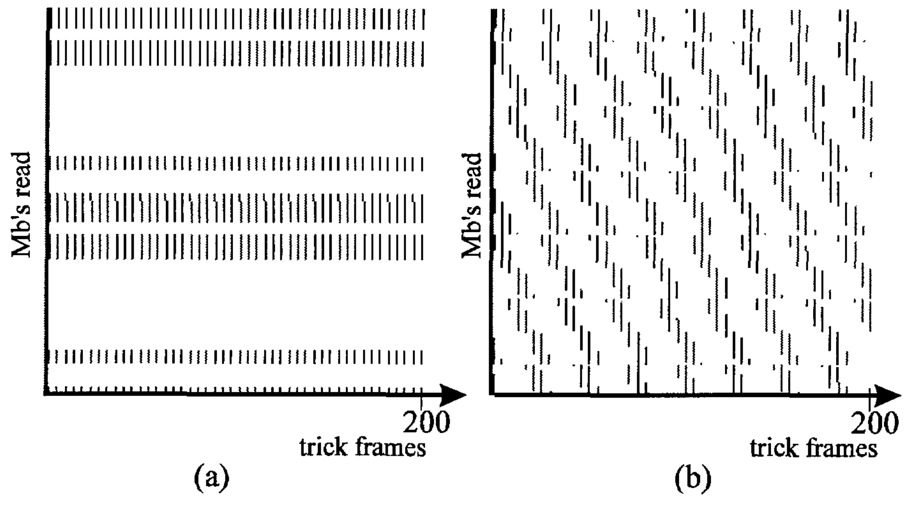


Figure 4.4: Model based macroblock updates for (a) n=3.0, (b) n=3.1 (Q=0.5, E=2, plain format, GOP13)

In Figure 4.4.a the resulting refresh pattern is given for the speedup of n=3.0. The figure illustrates that due to the periodicity of the read mask and the GOP bit stream, a periodic update pattern is formed. In the case of n=3.0 the picture update pattern is identical every time an update occurs; only two possibilities exist for the fast playback picture refreshes: a fast playback picture contains either no refreshes or it only contains refreshes for those macroblocks that were most recently refreshed. Visually this means that a certain part of the picture will never be updated and an other part of the picture will be updated for every intra coded picture in the bit stream.

In Figure 4.4.b the refresh pattern for n=3.1 is shown. For this speed it can be seen that, due to the frequency shift of the read mask, a phase shift will exist for the recovered sections from consecutive pictures. The macroblock updates will thus cycle through the entire fast playback picture for consecutive pictures and each macroblock will be refreshed regularly.

The average picture age, measured over a large number of pictures, will only be independent upon the number of fast playback pictures considered if all macroblocks are updated at regular intervals (Figure 4.4.b). When consecutive updates are aligned with each other (Figure 4.4.a) then the average picture age will be large and dependent upon the number of pictures considered.
The average picture age as a function of the speedup for the GOP3 and GOP13 structures, using the contiguous formatting, was numerically determined and the results are shown in Figure 4.2 . The average was taken over 200 fast playback pictures. The peaks at n=3.0 and n=6.0, which occur for both GOP structures, can be explained analytically. At these speeds the same pattern is read from every group of 6 double tracks (Equation (4-2)). As the GOP size is a multiple of 6 tracks (Equation(4-1)) a fixed selection of data will be made from consecutive pictures. The same reasoning applies to n=4.0 for odd GOP lengths.
Most of the other peaks can be explained in more general terms as follows: given the GOP period T_I, a speedup with the period T_R=T_I will synchronize the read pattern in frequency with consecutive Intra pictures. The more general case is one where T_R fits an integer k amount of times in the GOP period T_I such that using Equations (4-1) and (4-2) the synchronizing frequencies can be calculated as follows:

$$k \cdot T_R = T_I$$

$$k \cdot 2 \cdot n \cdot L_T = 6 \cdot L \cdot L_T\,, \qquad \text{where } k \in N,\quad k \neq 0$$

$$n = \frac{3 \cdot L}{k} \tag{4-4}$$

It is important to note that, for a short GOP (e.g. GOP3) the period T_I will be small and only a limited number of synchronizing frequencies will exist. For longer GOP structures, more synchronizing frequencies will be applicable. This is confirmed by the plots of Figure 4.2; the GOP3 plot only has a few significant peaks, the GOP13 plot has many smaller peaks.
In Figure 4.5 a histogram is shown of the synchronizing frequencies calculated using Equation (4-4) for the GOP sizes $2 \leq L \leq 14$. This range for L was chosen because typically a GOP will stay within this range. Many of the peaks of the average picture age of Figure 4.2 can be found in this histogram. We can conclude that the synchronizing frequencies as determined by Equation (4-4) are a major cause for the average picture age peaks of Figure 4.2.

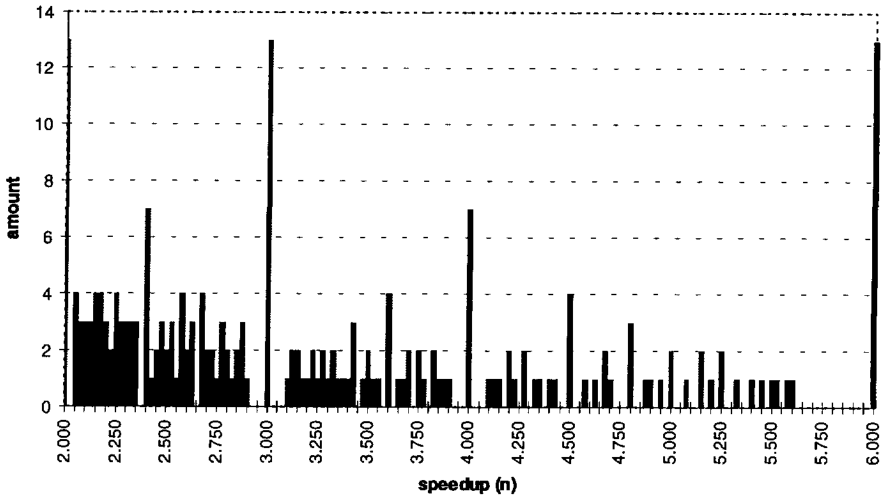


Figure 4.5: Histogram of synchronizing speedups for GOP size $2 \leq L \leq 14$. (resolution 0.025)

It should be noted that, due to the pattern of the read mask within the section of n tracks, it is likely that not all synchronizing frequencies have been found. Moreover, Figure 4.5 does not give a feeling for the severity of the problem at the various speeds.

In order to numerically identify all synchronizing frequencies, the average picture age is simulated for all GOP lengths between L=2 and L=14. For a certain format (plain or contiguous) this results in a set of 13 plots like the ones in Figure 4.2. The results for the plain format are summarized in Figure 4.6. In the figure a threshold is applied to the average age plots and for each speed the number of peaks larger than the threshold are plotted. In two dimensions the graph can be compared to Figure 4.5. On the third axis different threshold values are taken such that the severity of the peaks can be evaluated. For the contiguous format an equivalent plot can be formed.

The results for the plain and the contiguous format are equivalent to each other and based on the simulations no preference for one or the other tape format can be made. As an example Figure 4.7 shows the plot of the average picture age for the GOP13 for both the plain and the contiguous format. The figure shows that many of the synchronizing frequency peaks are common to both formats.

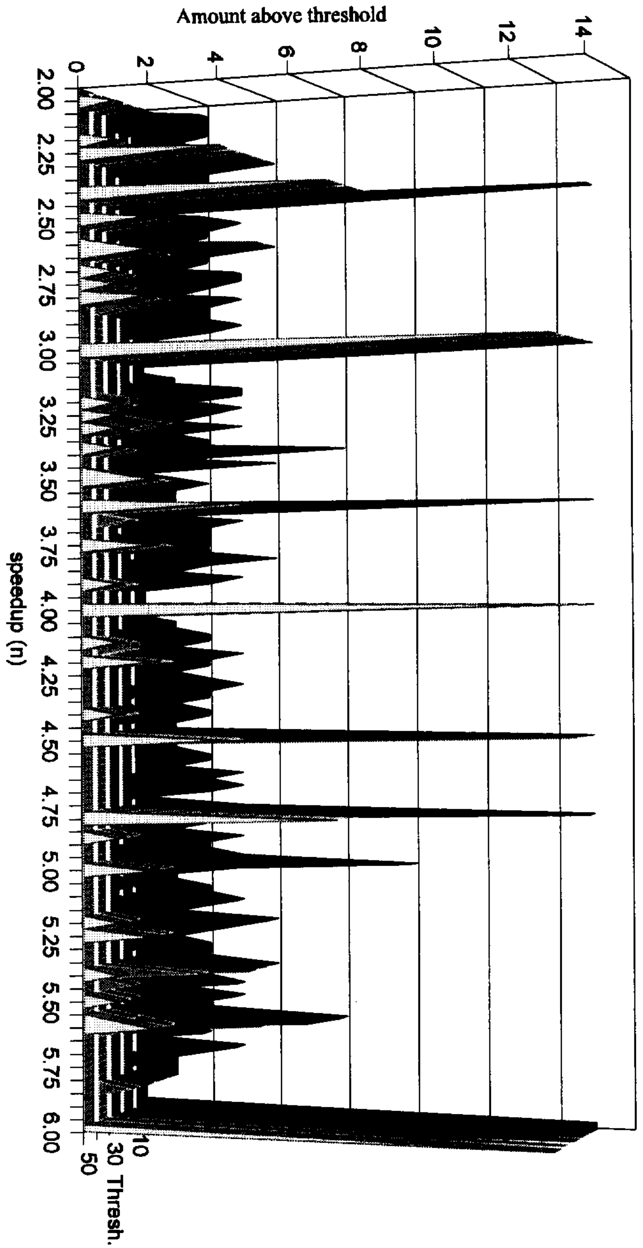


Figure 4.6: Amount above threshold for plain format

The few differences are caused by the specific read mask pattern for the respective formats. Referring to Chapter 3 an entirely different read pattern exists for the contiguous format within a DFS; Equation (4-2) is therefore not entirely applicable to the contiguous formatting. The read pattern within a DFS will however only have a minor effect on the synchronizing frequencies and for all practical purposes the synchronizing frequencies will be the same for both formats.

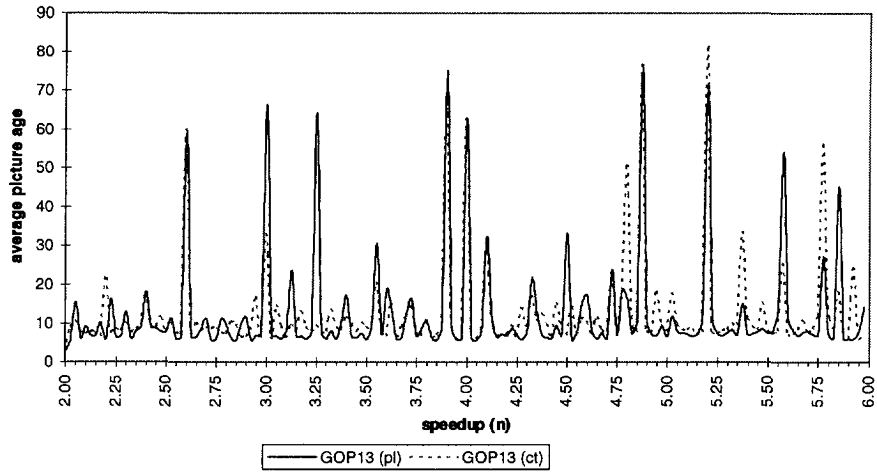


Figure 4.7: Comparison of the average picture age for the plain and contiguous format for GOP13.

One additional point is apparent in Figure 4.7. If we disregard the peaks, the average picture age does not increase or decrease as a function of the speedup factor. For every fast playback picture, data is collected during a period T_T, which is proportional to the speedup n. During this period only a fraction f_{READ}, which is inversely proportional to the speedup n, of the data is recovered. The model thus predicts that the average amount of usable data recovered is independent upon the speedup factor.

4.3.4 Solution to Periodicity of updates

We will discuss two methods to alleviate the problem of the periodic updates of the same picture sections; avoiding the speed-up factors where problems occur and performing a time variant mapping to assure that different picture sections are updated even for the speed-up factors where initially problems occurred.

4.3.4.1 Non-integer Speedup Factors

An obvious solution to the problem caused by the periodicity of the signals, is to operate the recorder at a specific speed-up factor where the periodicity is of no concern. The speed-up factors to be avoided for the model bit stream, i.e. the taboo speeds (n_p), can be determined from Figure 4.6 . Those speeds, where for a high threshold, large peaks exist are definitely taboo. Alternatively, the speeds where even for lower thresholds small or no peaks exits are attractive speedup factors. For instance it can be seen that n_p=3.0, n_p=4.0 and n_p=6.0 are disadvantageous. The respective alternate speeds, n=2.9, n=3.9 and n=5.9, are expected to yield better results.

A remaining question is what margin should be kept around the taboo speeds, such that the acceptable speed-up factors can be determined. To some extent this is dependent upon the Q-factor, but in general model based experiments have shown that, for the above speeds, the problem of the periodicity will only be of concern for those speeds that lie within the interval:

$$n_p - 0.1 < n < n_p + 0.1 \tag{4-5}$$

Any speedup outside of this interval will be acceptable.

Having determined the taboo speeds for any GOP length $2 \le L \le 14$ we also know the taboo speeds for sequences with changing GOP lengths.

4.3.4.2 Time Variant Mapping

An important question is whether or not it is possible to design a mapping such that there will be no taboo speeds. A possible approach which alleviates the taboo speed requirements is the time variant mapping. With a time variant mapping consecutive DFSes are mapped differently and no synchronizing frequencies can occur. There are two possible approaches to such a time variant mapping.

Firstly the necessary stuffing packets, required to fill the DFS from the video bit rate of 10 Mb/s to the 12.5Mb/s of the recorder, can be formatted differently for consecutive DFS's. By inserting the dummy packets area at random positions a forced fluctuation in the GOP tape position can be introduced such that the problem of the synchronizing periodicities can be alleviated.

Secondly, the map for each DFS can perform a contiguous re-mapping with a random offset. As is the case for the first solution, the position of the burst within the DFS will be modulated randomly. The advantage of this second approach is that a larger fluctuation of the GOP start position is possible. For this second approach, the appropriate map must be recovered on playback, based on the data received. The re-mapping address must thus form part of the recorded signal and cannot simply be based upon the sync block number.

For both time variant approaches the complexity of the recording adaptation interface is increased to support time-variant function. This increased complexity is outside of the scope of the hardware developed for the verification model. Moreover, in practice we feel that as soon as complex operations need to be performed on recording, the full dedicated fast playback solution should be preferred. Therefore, we will not make any use of the time-variant mapping and limit ourselves to avoiding the taboo speeds.

4.3.5 Exploiting Periodicity

The periodic read mask can also be exploited to enhance the rudimentary fast playback picture. It is possible to force the Intra pictures bit stream positioning to approach the bit stream model, either by employing a stringent rate control or by inserting the dummy packets at the appropriate locations. With such a known positioning of the Intra slice information a whole new basis for the fast playback problem exists.

Given a known Q-factor, a speedup can be chosen such that the picture age is minimized; for certain speeds a maximal amount of old macroblocks will be updated. Obviously the individual macroblock sizes are still uncertain but on the average the speed chosen for the model will have the best performance.

Like the hybrid solution of Section 4.2.3, this enhanced rudimentary fast playback moves us from the rudimentary fast playback principle towards that of the dedicated fast playback streams. The complexity of the recording adaptation interface is increased and is expected to be comparable to that of the dedicated fast playback stream solution. As the dedicated fast playback solution will yield more controlled and better visual results the enhanced rudimentary trick modes will not be pursued.

4.4 Simulation results

In the previous section we have performed a model based analysis of the update mechanism for the rudimentary fast playback. A model of the bit stream was used to predict the taboo speeds. In this section we will evaluate these taboo speeds for true video, where some variation will exist in the length of the consecutive GOPs.

4.4.1 Periodicity for coded MPEG

For loose rate control, the error in Equation (4-1) and the different usage of bits per macroblock will cause a statistical deviation from the worst case model of the previous section. In this section we will evaluate a concrete case where the GOP size distribution is governed by the rate control of the MPEG test model 4 [MPEGTM4].

Figure 4.8 shows the relative GOP size distribution evaluated over 1200 pictures from the "123" Alberville Olympics sequence. 0% signifies the nominal GOP size as predicted by the model for a certain bit rate. The results are obtained for a bit rate of 9.6 Mb/s. The figure shows that for both GOP sizes, the relative error stays within ±5% with a high probability. The standard deviation of both distributions is about 3%.

The parameter of importance is the standard deviation of the absolute GOP size. For the GOP3, the standard deviation is equal to half a track. This means that the varying GOP sizes will cause only a small perturbation from the model of Section 4.3. For the GOP13 the standard deviation of the absolute GOP size will be equal to 2.5 tracks. This deviation is more significant and it will have a randomizing effect.

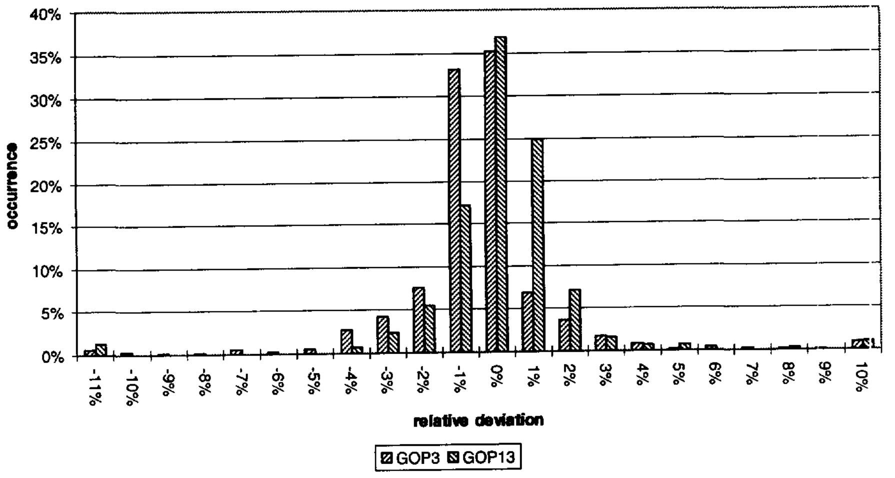


Figure 4.8: GOP size relative deviation from the nominal for L=3 (GOP3) and L=13 (GOP13).

ideal age 1.00	plain		contiguous	
	average	*sdev*	*average*	*sdev*
n=3.0	13.80	6.03	6.71	3.64
n=3.1	1.60	0.28	1.29	0.49

Table 4.1: Picture age for n=3.0 and n=3.1 for GOP3

Table 4.1 summarizes the picture age distributions (average and standard deviation) for four experiments with fast playback sequences of 200 pictures encoded with a GOP length of L=3. The ideal picture age, where all intra encoded data is recovered, is determined as the ratio between the speed-up n and the GOP length L (which in this case is n/L=1.0). For both tape formats there is a large difference between the average picture age for n=3.0 and for n=3.1. The standard deviation in the GOP length for GOP3 thus is not enough to alleviate the problem of the synchronizing frequencies and n=3.0 is still a taboo speed. Therefore it is required to operate at the speed of n=3.1 instead.

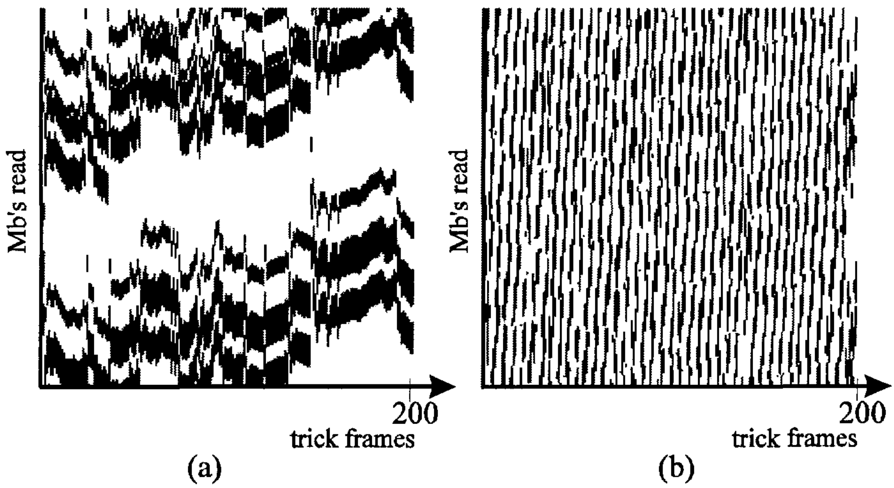


Figure 4.9: Fast playback updates for (a) n=3.0, (b) n=3.1 (Q=0.5, E=2, plain format, GOP3)

Figure 4.9 shows the update pattern that yields the results of Table 4.1 for the plain format. This figure confirms that, for this small GOP size, the taboo speed of n=3.0 is indeed a problem, even though the varying GOP size introduced a statistical variation. We can therefore confirm the model prediction of the synchronizing frequencies, even for a loose rate control.

ideal age	**plain**		**contiguous**	
4.33	*average*	*sdev*	*average*	*sdev*
n=3.0	13.71	4.74	8.12	2.03
n=3.1	11.32	3.15	7.17	1.83

Table 4.2: Picture age for n=3.0 and n=3.1 for GOP13

The average picture age for both the plain and the contiguous format for the longer GOP13 are given in Table 4.2. For this longer GOP the ideal age has increased to n/L=4.33 . The simulation results show that when this particular rate control is applied to longer GOPs the taboo speeds become less relevant. For n=3.0, there will be little synchronization with the read mask such that all picture sections will be updated. The large standard deviation in the GOP size assures that there is enough fluctuation in what sections of the Intra picture will be read.

4.4.2 Influence of Tape Format on Picture Age

From Tables 4.1 and 4.2 we see that, contrary to the results from the model based analysis, the average picture age is lower for the contiguous format than for the plain format. We know that for both formats the same amount of information (sync

blocks) is recovered from the tape. The difference between the two formats is that for the contiguous format long burst of data are recovered as opposed to many short burst for the plain format.
Since the data will only be useful after the first slice *start_code*, All the data before the start code will be discarded. For every single burst a certain amount of recovered data will thus be useless such that long bursts will have an advantage over short bursts. This synchronization time explains the observed difference between the plain and the contiguous format.

4.4.3 Neighborhood Integrity

4.4.3.1 Visual Evaluation of Fast Playback Picture

At this point we have minimized the picture age by avoiding the taboo speeds and we have observed that the contiguous format has an advantage over the plain format. An important question is what the effect of the formatting is on the visual quality of the picture.
Figures 4.10 and 4.11 give an example of a single fast playback picture for the plain and contiguous format, respectively. The example is the first picture after a scene change. Both pictures have a comparable picture age but a visually quit a different quality can be observed. With the plain format several burst will cause several separate updates resulting in a picture where relatively small sections originate in the same normal play picture. With the contiguous format only one large update has taken place such that the top half of the picture still represents the old scene and the bottom half represents the new scene. In general, one large update is visually much more attractive than many small updates.

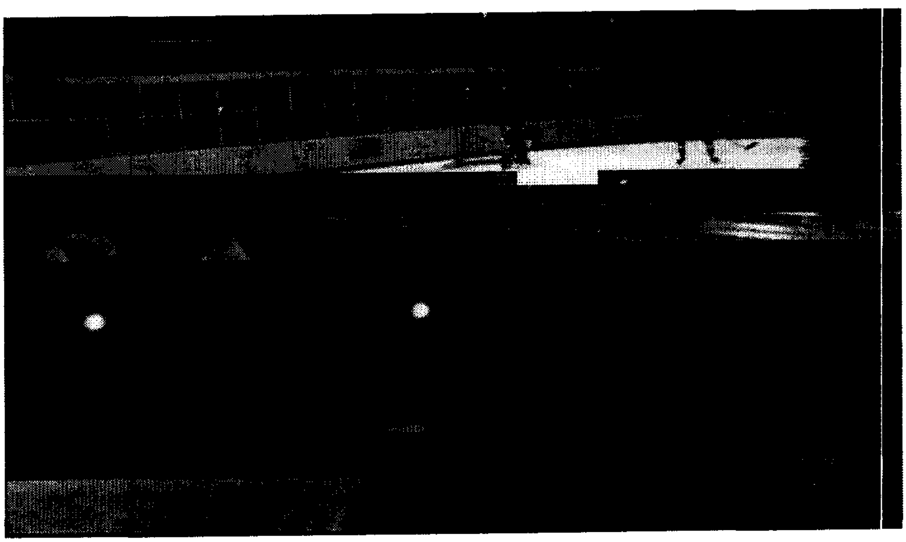


Figure 4.10: Picture after scene change, using plain format

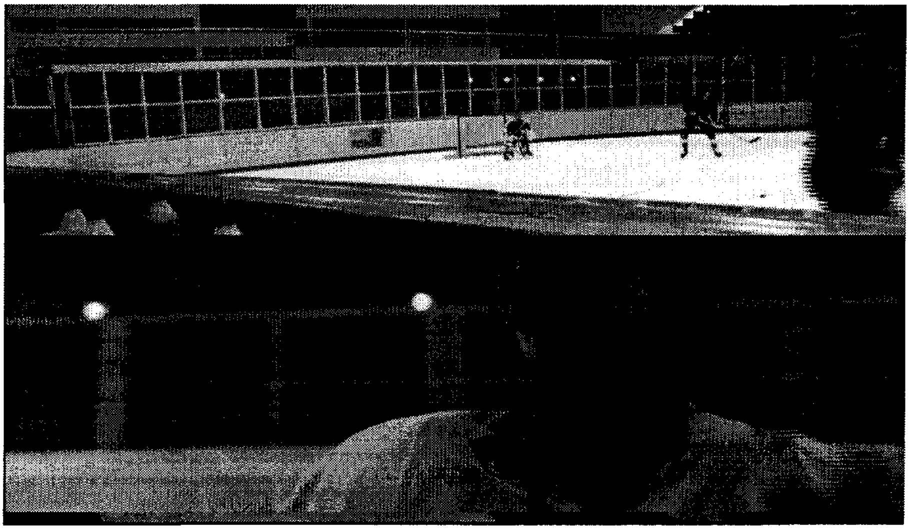
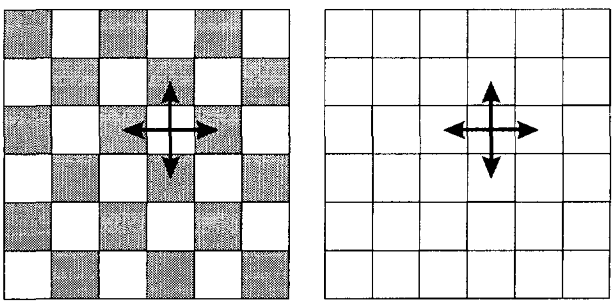


Figure 4.11: Picture after scene change, using contiguous format

Moreover, as the macroblocks follow each other in a scan wise order, a small update will form a stripe, only a few macroblocks high, across the whole picture. Such a stripe, which gives the effect of looking through a letterbox, yields less interpretable information that a block of equal height and width would.

4.4.3.2 Evaluation

To quantitatively identify the degree in which the formatting influences how macroblocks from different normal play pictures end up as neighbors in fast playback pictures, we define a neighborhood integrity (N_i) measure. This measure quantifies how many neighboring macroblocks originate in the same normal play picture.



(a) (b)

Figure 4.12: Neighborhood integrity

Figure 4.12 shows two extremes on the neighborhood integrity scale. In Figure 4.12.a the shaded macroblocks originate in an older normal play picture than the white macroblocks. For this worst case, where no two neighboring macroblocks (four connected) originate in the same normal play picture, we define $N_i=0$. In Figure 4.12.b all macroblocks have been refreshed at the same time and they all originate in the same normal play picture; for this situation we define $N_i=1$.
For an arbitrary fast playback picture an average neighborhood integrity is calculated by considering all boundaries between neighboring macroblock pairs. Boundaries between blocks of different origin contribute *zero* to the average, boundaries between blocks of equal origin contribute *one* to the average.

The average neighborhood integrity was measured for the 200 simulated fast playback pictures of the Alberville "123" sequence for n=3.1. Table 4.3 summarizes the results of the experiments. The experiments numerically confirms that the contiguous format offers a clear advantage over the plain format.

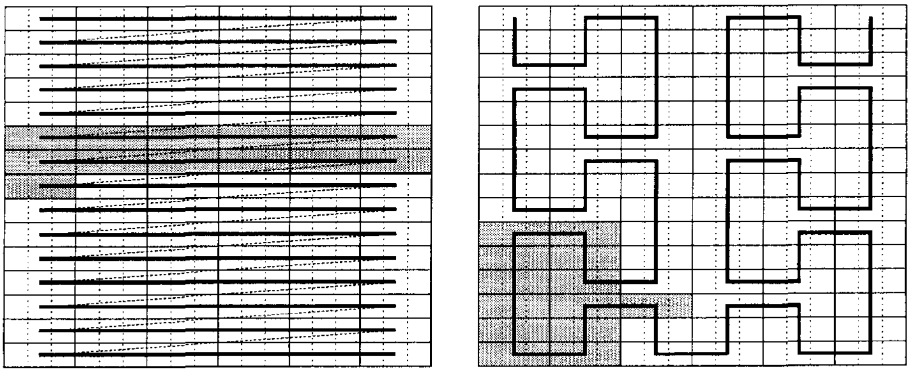
	Plain		Contiguous	
	average	sdev	average	sdev
GOP3	0.882	0.059	0.951	0.030
GOP13	0.905	0.060	0.962	0.025

Table 4.3: Neighborhood integrity

4.4.3.3 Maximizing the Neighborhood Integrity

The observation that a "letter-box" update is less advantageous than a block update of an equal amount of macroblocks is significant in the context of MPEG encoding. Within the MPEG standard a scan wise ordering of the macroblocks and the slices is prescribed. As such, a picture update will always form a stripe of a certain number of macroblocks. It would be favorable to order the macroblocks differently in the original stream such that an update is more spatially localized and forms a block of new image data instead of a stripe.
Within the specification of the standard it is not allowed to order the macroblocks differently. However, every slice, and every macroblock within a slice, has a unique address such that technically speaking a different ordering is a possibility within the standard. We will briefly step out of the scope of the standard to discuss such a different ordering.

If the size of the Intra slices is limited to a few macroblocks then a possible approach to accomplish a different order of the macroblocks would be to re-order the slices within the Intra picture. This slice ordering can be such that the macroblock chain forms a space-filling curve, within the limitation of the chosen slice size. An example of such an ordering with very short slices of 3 macroblocks each is given in Figure 4.13.b; Figure 4.13.a gives the standard slice ordering as a reference. The shaded area indicates a possible update area when 13 consecutive slices are updated.



(a) (b)

Figure 4.13: Slice ordering (a) standard, (b) space-filling curve (3 macroblocks per slice).

Experiments have shown that the usage of this alternate macroblock scanning would indeed improve the objective neighborhood integrity and the subjective visual quality. The improvement is dependent upon the burst sizes. For the plain format the neighborhood integrity can be increased by 10%, for the contiguous format only 2% to 3% is possible. We will not pursue this alternate scanning because it is outside the scope of the MPEG standard.

4.5 Bit stream Validation

4.5.1 Approach

We now discuss the hardware requirements for decoding a rudimentary fast playback signal and define a hardware demonstrator to perform the real-time rudimentary fast playback.

In a practical recorder that carries out rudimentary fast playback, the useful intra-slice information obtained from tape must be compiled into a valid MPEG bit stream, so that a normal MPEG decoder is able to interpret and decode this data. It is therefore left to the playback adaptation interface to reconstruct a valid transport/video bit stream.

One of the main questions is what the "standard" decoder is capable off. This will determine the necessary adaptation interface functionality. The PES fast playback byte (Section 2.5.2) allows for the signaling of an incomplete slice structure, i.e. the signaling that pictures are only partially filled with usable data and the data for the non-recovered slices should be copied from previously decoded pictures. When a decoder supports the special DSM provisions (Section 2.5.2) including incomplete slice structures, then the function of the adaptation interface is as follows:

- depacketize the recovered packets (sync-blocks)
- extract the intra coded *Slices*
- group these *Slices* in *Picture* presentation units, i.e. add virtual timing
- Add *Picture* headers
- Form a PES
- packetize the *Pictures*

At this point it is not clear whether the DSM provisions will generally be supported. We know that some of the currently available experimental decoders do not support incomplete slice structures [ST3500]. As such, for the demonstrator and possibly for all future systems there is an important requirement of complete slice structures. The adaptation interface needs to reconstruct a video stream that is decodable by any standard decoder. The bit stream thus needs to satisfy the restricted slice structure, where the missing slices are filled with previously received data.

4.5.2 Bit Stream Validator

Due to the requirement of the available decoders of a fully MPEG compliant input stream with a complete slice structure, the playback adaptation interface of the demonstrator will need to contain a bit stream validator. The role of such a validator is to take the invalid MPEG-video input bit stream and to generate a valid MPEG bit stream decodable by any standard decoder.

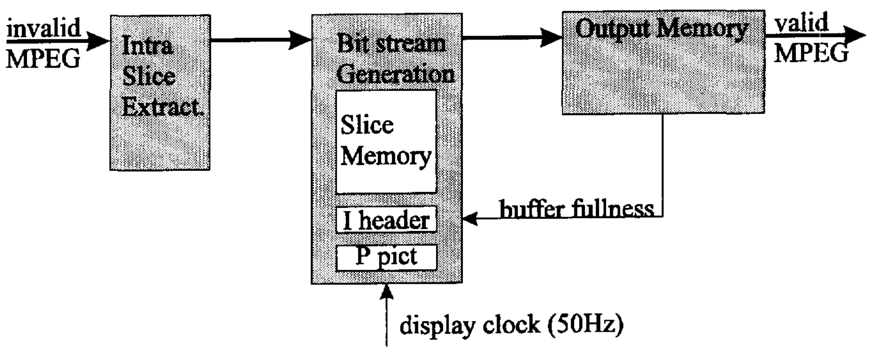


Figure 4.14: Bit stream validator architecture

Figure 4.14 shows the system architecture of the validator that can be used for both fast forward and reverse. As the MPEG *start_codes* are byte aligned the entire system is byte oriented. The validator contains three major elements:

- the Intra slice extractor
- the bit stream generator
- the output memory

The Intra slice extractor has the simple function of searching through the received invalid MPEG bit stream for intra coded slices. When these intra coded slices are found then the slice position can be deduced by decoding the slice vertical position and the first macroblock address; consequently the slice can be passed to the bit stream generator.
The bit stream generator stores the most recent version of the bit stream section of every slice in a memory. At any time the memory contains slice data for the whole picture and an intra coded picture, that starts with a pre-stored Intra-picture header and is followed by slices, can be generated. From one output picture to the next only those slices which have been changed by a recent update will be new; for the other slices the same data will be re-transmitted.
The time at which the pictures are generated is governed by the picture frame rate; there must be a picture for each frame period. The output bit stream must be VBV compliant, i.e. the underflow and overflow constraints discussed in Section 2.3.9. must be satisfied. This can be controlled as a simple encoder would do it, by monitoring the buffer fullness of the output buffer. The bit stream generator can, however, not perform a rate control by adapting the quantizer. It thus performs a rate control by deciding which of the following should be the next picture:

1. an intra coded picture with actual data
2. a small dummy predicted picture.

The dummy predicted picture is a fixed bit pattern that tells the decoder that each macroblock is simply copied from the previous reference picture (no motion, not coded). It has as great advantage that it only requires a small number of bits. The ratio of the dummy P pictures to the actual I pictures is governed by the bit rate chosen for the resulting MPEG video stream.
The output buffer size must be equal to the size of the input buffer of the decoder. This is governed by the profile and level of the decoder under consideration; for the *Main profile*, *Main level* this will be slightly under 2Mbit.

Finally the timing of the whole system is of importance. In an actual system that makes use of the transport stream the 50Hz presentation unit (display) clock could be added to the signal by the inclusion of presentation time stamps. In the demonstrator, where there is no means to include time stamps, the synchronization of the decoder display clock to the validator display clock is achieved by locking the clocks to each other. Therefore, in accordance with the approach for the verification model (Section 2.5.3) all display clocks are locked to each other through the channel bit rate.

4.5.3 Macroblock Based Extension to the Validator

The described bit stream validator has one drawback due to its slice oriented operation. The last slice of a burst of slices will never be complete; it will be broken off by an error code. The reconstructed picture in this area is thus not controlled by the validator but depends upon the concealment capabilities of the decoder.

Furthermore, the macroblock oriented slice recovery implies that a fixed slice structure is used. Any particular slice structure can be supported, but when the slice structure changes from one picture to the next the merging of slices that originate in different pictures is expected to cause problems.
The above slice problems can be solved by performing the bit stream validation on a macroblock basis. When a macroblock is recovered it is stored in the validator memory. The original slice structure is lost and the generated bit stream employs a pre-set default slice structure into which the macroblocks are concatenated.
The main problem with this macroblock oriented approach is that macroblocks are not byte aligned. Furthermore, the only way individual macroblocks can be parsed from the bit stream is by VLC decoding all the preceding codewords, i.e. unlike slices there is no unique start code in the header of every macroblock. The complexity of an intra macroblock extractor thus is much higher than that of an Intra slice extractor. The bit stream generator complexity on the other hand is about the same, with the same memory requirements.

5. Fast playback stream extraction

5.1 Introduction

In Chapter 4 we have shown that rudimentary fast playback is a feasible solution but that some restrictions exist. In particular, the drawback of rudimentary fast playback is the strong dependence of the fast playback picture quality on the way the normal play picture was encoded. Depending on the scanner to tape phase, the fast playback video will be different every time fast playback is performed. Furthermore, where the recording adaptation interface will be extremely simple, the playback adaptation interface will need to incorporate the function of a bit stream validator.

In Section 3.3.2 the idea of using a dedicated bit stream for fast playback was introduced. Such a dedicated stream must be recorded in addition to the normal play signal, using the spare capacity which is assumed to be available on the recorder. The advantage of this dedicated stream fast playback is that the resulting picture quality is pre-determined and independent upon the scanner to tape phase.

The format design of multiple copies of this dedicated stream was developed in Section 3.5. Using this design method we can determine how many copies of the dedicated stream are required and, given a certain spare capacity, the available bandwidth for the dedicated fast playback stream can be determined. The typical system under consideration in this thesis is the recording of a normal play bit stream of 10Mb/s on a M4 helical scan recorder with the effective bit rate 12.5Mb/s, yielding 2.5Mb/s spare capacity. This spare capacity will need to be partitioned over the different dedicated fast playback streams for the different desired speedup factors n. The results of Section 3.5 show that for a speedup factor n we will typically need to record n or n+1 copies (E=2 and Q=0.5) of the same dedicated fast playback stream. When, for example, we use the spare capacity for two dedicated fast playback streams this yields an available bandwidth of about 1 Mb/s per dedicated stream.

The remaining question now is what the content of the dedicated stream should be and how this stream can be transcoded from the normal play stream. A particular form of transcoding is that of bit stream extraction, where the fast playback stream is based on codewords copied from the original stream. The problem of dedicated stream extraction from an MPEG encoded normal play stream is not of sole interest to fast playback with helical scan recorders. In fact it is expected that many serial recording media with limited possibilities for random access will only be able to support an attractive fast playback feature by the use of a low bit rate dedicated bit stream.

The fast playback stream must constitute a valid MPEG bit stream which can be decoded by the same decoder as used for the normal video. We will consider *main* *profile* at *main level* decoders. To ensure decodability by the same hardware, the frame rate of the fast playback stream should be no more than that of the normal stream. As the fast playback stream constitutes a storage overhead, its bit rate should be significantly lower than that of the normal play stream. The bit rate of 1 Mb/s suggested for the dedicated fast playback stream is only a typical example; in our discussion an wider range of target bit rates will be considered.

In this chapter several fast playback bit stream extraction solutions are presented and evaluated with respect to their performance and implementation complexity. The conceptually simplest solution for the transcoding problem would be to completely decode the normal video stream and encode a low resolution, temporally sub-sampled version at a low bit rate. This requires the presence of a decoder, an encoder and a significant amount of memory in the recording adaptation interface. Our objective is to achieve a lower complexity by extracting a fast playback bit stream without fully decoding the original bit stream, i.e. the coefficients of the original stream are not re-quantized and instead a selection of the ac coefficients is made for the fast playback stream.
In Section 5.2 various approaches to fast playback stream extraction and their basic architectures will be discussed. In Section 5.3 the possibilities for the coefficient selection are discussed. The effect of the rate control will be deferred to Section 0 such that the performance bounds of the proposed methods can be evaluated. In Section 5.5 the different solutions will be evaluated for a typical video sequence and the trade-off between the different solutions is discussed.

5.2 Approach to Transcoding

5.2.1 Usage of Intra Pictures

In extracting a fast playback stream only the intra frame information of the original stream is used. The reason for this is that, equivalently to the rudimentary fast playback problem, only a fraction of the original frames can be included in the fast playback stream. The references needed when including predicted frames may therefore not be available or may have been changed by the extraction procedure.
Figure 5.1 gives an example of how intra frames are extracted for a fast playback bit stream with a speed-up factor n=3. The figure presents two possibilities for the fast playback sequence. In Figure 5.1.a the resulting fast playback bit stream has a frame rate that is equal to that of the original stream. This is achieved by stuffing small (~2.6kbit) dummy P pictures (shaded) between the extracted I pictures. The dummy P pictures only contain *not-coded* macroblocks and are thus given by a fixed bit pattern. This reduces the frame rate of the fast playback video sequence. In Figure 5.1.b the resulting bit stream has a reduced (halved) frame rate and in principle contains only the extracted intra coded pictures. Note that in order to guarantee the prescribed frame rate, the fast playback stream may occasionally contain a dummy P picture when no intra refresh has occurred. Also, depending on

the original Group of Picture (GOP) structure, not all intra coded frames may be used.

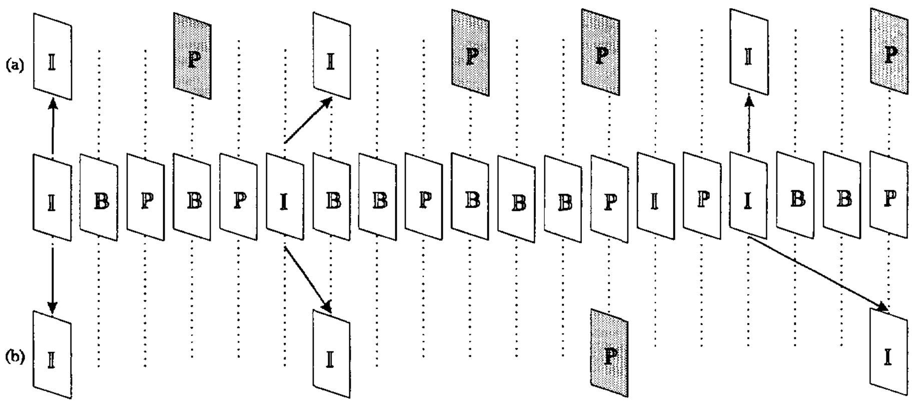


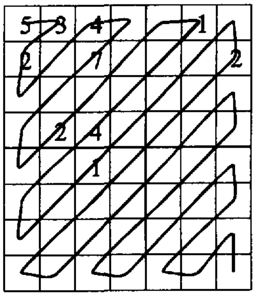
Figure 5.1: Two possibilities for fast playback stream extraction

For a constant GOP size the two solutions are equivalent with respect to the refresh rate of the intra coded pictures. In our evaluation we will use the second approach, where the fast playback frame period is an integer multiple (k_m) of the original frame period. When generating a video bit stream for a speed-up of n, the fast playback video stream will have a frame for every $n*k_m$ frames of the normal stream. We assume that the frame rate reduction is such that every cycle of $n*k_m$ contains at least one GOP and no dummy P pictures will be required.
A special case exists for the "Design for Two Speed-up Factors with Single Fast Playback Stream" approach (Section 3.5.2.4). In this case k_m will be different for the different playback speeds as $n*k_m$ is a constant, determined on extraction.

5.2.2 Transcoding of Intra Pictures

The desired bit rate for the fast playback stream is significantly lower than the rate for the normal playback stream. If the entire intra pictures are used for the fast playback bit stream then a very large frame period multiple (k_m), possibly as large as 15, will be required (low frame rate) to compensate for the high bit rate of the intra coded frames. To achieve an acceptable frame rate, the intra pictures of the fast playback stream should be transcoded to a lower bit rate. Nevertheless, there exists a trade-off between the frame rate and the picture quality. In a practical system a frame rate will be chosen such that an acceptable picture quality is possible.
The extracted stream needs to be MPEG compliant. It will thus include all headers, up to and including the block DC coefficients. These headers can be copied directly from the original sequence. The bit rate entry and the frame rate factor entry in the sequence header are, however, modified to the new situation.

N*N block



coefficient domain (cd)

tuples (r,l)

(0,5), (0,3), (0,2), (3,4),
(1,7), (3,2), (3,1), (2,4),
(4,1), (4,2)

run-length domain

VLC codewords

001001100
001010
01000
00000000100110
00000010100
001001000
001110
0000000101000
001100
00000011110
10

bit domain (bd)

Figure 5.2: Block representation domains

For the discussion on how to obtain the block coefficient codewords for the fast playback stream, a clear understanding of the coefficient representation in the bit stream is necessary. In Figure 5.2 the different domains in which a quantized DCT block can be represented are shown. In the coefficient domain a block contains N*N (N=8) integer entries that correspond with the quantized DCT coefficients. Many of the entries will be zero. This is particularly the case for those entries that correspond with the high frequencies as these are quantized more coarsely (visual weighting) by the encoder. In the run-length domain, the non-zero coefficients are re-ordered in a zigzag scan manner and they are represented by a run-length tuple (r,l), where the run (r) is equal to the number of zeros preceding a certain coefficient and the level (l) is equal to the value of the coefficient. The transition from the coefficient domain to the tuple domain will be called *tuple coding*. In the bit domain the (r,l) tuples are represented by variable length coded (VLC) codewords. The codewords for a single block are followed by an end of block (EOB).

A wide range of approaches to transcode the normal stream to a fast playback stream are possible. Figure 5.3.a shows a straightforward transcoding approach for a single block. The codewords in the bit domain (bd) are variable length decoded (VLD), yielding run-level (r,l) tuples. The tuple decoding (TD) yields quantized coefficients in the coefficient domain (cd). These coefficients are de-quantized (DQ) to obtain the DCT coefficients (dctd). From this point the encoding can be done as in a normal intra encoder, namely quantizing (Q) subject to a rate control, tuple encoding (TC), and VLC encoding, yielding the new codewords. We will not elaborate on this *full transcoding* approach but we will use it as a final performance reference requiring a relatively high complexity.

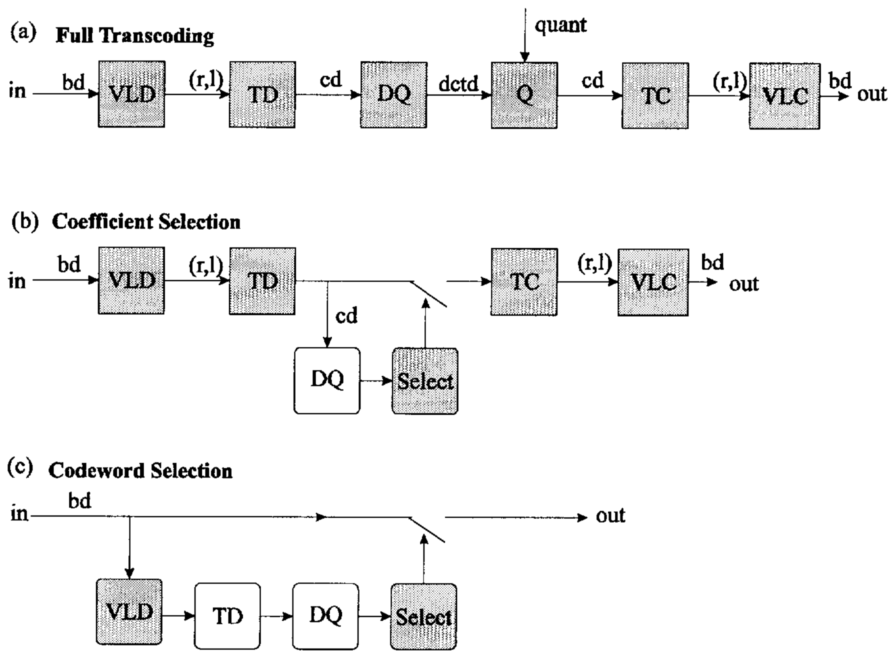


Figure 5.3: Possible block transcoding architectures for intra pictures

Figure 5.3.b shows an architecture with a decreased complexity, in which the blocks from the original bit stream are decoded up to the coefficient domain. In the coefficient domain a selection is made of those coefficients that need to be retained for the corresponding block of the fast playback bit stream. The resulting block is re-encoded from the coefficient domain down to the bit domain, i.e. tuple coding and VLC encoding. Note that the selected coefficients are *not re-quantized.* In this approach the problem is to decide which coefficients from the coefficient domain to use for the fast playback picture to achieve a good overall picture quality at the required bit rate. The *coefficient selection* mechanism must therefore be governed by a rate control mechanism. For many selection mechanisms the coefficients will still need to be de-quantized to determine their contribution to the picture quality.

In the approach shown in Figure 5.3.c, an even lower complexity is achieved by performing a *codeword selection* in the bit domain. The selected codewords will simply be copied and this does not involve the decoding up to the coefficient domain and re-coded to the bit domain. Given the run-length encoding, this is equivalent to a limited freedom coefficient selection where only a number of consecutive coefficients of the block will be used for the fast playback stream. For a certain block we must start with the first coefficient and use a certain number of consecutive coefficients. We will denote the number of coefficients selected (c) as the cut-off level.

The *codeword selection* mechanism will, at the very least, require the VLD decoding to have taken place to identify the individual codewords. The original blocks will be decoded up to the run-level domain where a number of consecutive tuples will be selected, i.e. the cut-off level will be determined. From this the number of bits from the bit domain to be used for the intra frame can be determined. As is the case for the previous approach, some selection mechanisms may require tuple decoding and even de-quantization in order to perform an appropriate selection.
The *zonal codeword selection* method as suggested in [Lane93, Yana93b, Okam93] has the lowest hardware complexity. The architecture of the zonal codeword selection method is also given by Figure 5.3.c. However, in contrast with the codeword selection, it has a fixed cut-off level which is not determined by image content (the represented values) and only depends upon the rate control.

In the discussion and evaluation of the different solutions, we will first consider the coefficient/codeword selection separately from the rate control such that the optimal performance can be evaluated. The constraint of controlling the bit rate for the individual picture will be added later.

5.2.3 Relation to the Data Partitioning Scalable MPEG syntax

The *codeword selection* approach and in particular the zonal codeword selection approach, has a lot in common with the data-partitioning scalable MPEG syntax [MPEG2V]. In the data-partitioning scalable solution a high priority (HP) layer is formed by extracting all the header information, the DC codewords and a few AC codewords from the non-scalable stream. All remaining codewords go into the low priority (LP) layer. The number of codewords selected for the HP layer is fixed per slice and indicated by a cut-off parameter.
Several aspects of the scalable data-partitioning solution make it less appropriate for the fast playback stream extraction:

- If the received *main profile/main level* bit stream is converted to a data partitioned scalable stream then a scalable decoder will be necessary to decode the normal play bit stream while this was not the case for the bit stream prior to recording.
- The requirement of a fixed cut-off level per slice is too limiting and we will show that the selection of a different number of coefficients per block yields better quality.
- The dedicated fast playback stream must be decodable by a non-scalable decoder, which is not the case for the HP layer of the data partitioning stream. The most important reason is that the data partitioning does not add the EOB (End Of Block) codeword to the HP layer.

For the above reasons we will not make use of scalable data partitioning and limit ourselves to the recording of two (normal and fast playback) or more independently decodable video streams.

5.3 Coefficient/Codeword Selection Mechanism

For both the *coefficient selection* and the *codeword selection* approaches (Figure 5.3.b and Figure 5.3.c) a mechanism is required to determine the optimal coefficient or codeword selection per picture block of the used intra coded pictures. In this section we formulate and solve this problem by an optimization approach. Obviously this optimization in itself will impose certain hardware requirements. It is thus quite possible that a non-optimal solution in the distortion sense, will be preferred. Both the optimal and the sub-optimal mechanism presented in this section will be used in Section 0 in system that includes a rate control.

5.3.1 Formal Problem Statement

The problem of the optimal rate constrained *coefficient selection* applicable to Figure 5.3.b is formulated as follows: if X_i is the input intra coded frame and X_o is the low bit rate output picture, minimize the distortion $D(X_i,X_o)$ between X_i and X_o subject to a total coding bit budget R_{budget} for X_o. The thus goal is to solve:

$$\min_{\forall X_o}[D(X_i,X_o)] \text{ subject to } R(X_o) \leq R_{budget} \tag{5-1}$$

The distortion measure used to obtain this optimum is the weighted mean-squared-error, i.e. the visual weighting used by the MPEG quantization of DCT coefficients is included in the measure.

For the *codeword selection* of Figure 5.3.c the same problem formulation is applicable with the additional constraint of a limited freedom of the coefficient selection.

Figure 5.4 shows a typical rate-distortion (R-D) plot [Berg71], where any point (R,D) on and above the convex line represents a possible operating point that corresponds with a different coefficient selection. The convex line connects those operation points that are optimal for either a particular rate or a particular distortion. The objective thus is to find the *coefficient/codeword* subset that yields an operation point on this convex curve and satisfies the rate constraint as close as possible.

5.3.2 Lagrange Multiplier

The constrained minimization problem (5-1) can be solved by minimizing the following unconstrained Lagrange function [Ramc94, Ever63, Shoh88]:

$$J(\lambda) = D(X_i,X_o) + \lambda \cdot R(X_o) \tag{5-2}$$

where $\lambda>0$ is the Lagrange multiplier that has to be chosen such that the condition:

$$R(X_o) \approx R_{budget} \tag{5-3}$$

is satisfied.

Referring to Figure 5.4, Equation (5-2) is the equation of a line with slope $-\lambda$, i.e. $D=-\lambda \cdot R+J(\lambda)$. For every line with the slope $-\lambda$ the Lagrangian cost, of all (R,D) operation points on the line, can be found at the distortion axis crossing $J(\lambda)$.

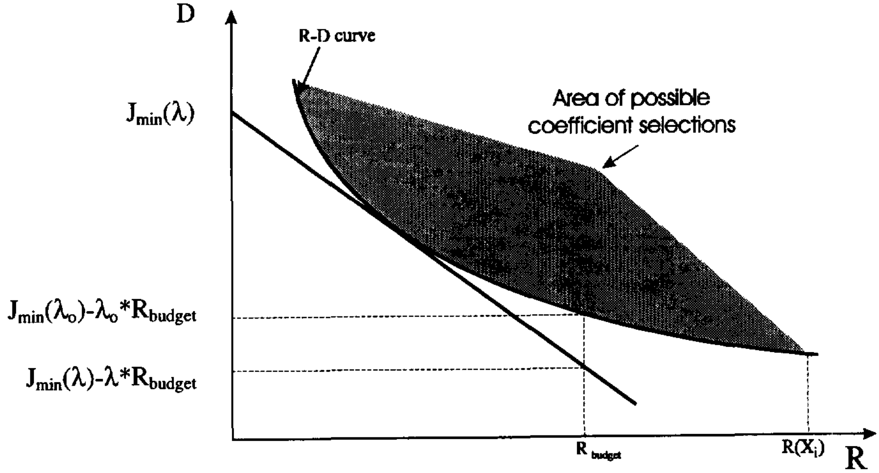


Figure 5.4: Rate-Distortion plot and optimal curve

The unconstrained problem is now given by the determination of that set of coefficients from a picture which result in the minimum total Lagrangian cost defined as:

$$J_{min}(\lambda) = \min_{(R,D)}[J(\lambda)] \tag{5-4}$$

It is intuitively clear that the minimization of the Lagrangian cost will be done by an operation point on the convex optimal curve at a point tangent to a line with slope λ.

It was shown in [Shoh88] that, for a quadratic distortion criterion D, the optimal coefficient search can be done independently for every 8*8 block for the fixed slope λ. In other words, given a certain λ, the minimum of Equation (5-4) is equal to the sum of the minimum for the individual blocks:

$$J_{min}(\lambda) = \sum_{b=0}^{\#blocks} \min_{(R,D)}[J_b(\lambda)] \tag{5-5}$$

The consequence is that for R-D optimality all blocks must operate at a point with a constant slope λ on their R-D curves, i.e. the slope $\delta D/\delta R$ of the R-D curve at the block's operation point must be constant for the entire picture. Intuitively this can be understood by the argument that if the operating point of different blocks correspond with a different slope λ, it is profitable to shift bits from a block with a shallow slope to a block with a steep slope; the total distortion will then decrease and thus the operating point was not optimal to begin with.

The advantage of the Lagrangian optimization is that, for a particular λ, Equation (5-2) can be minimized independently for each individual block and the total Lagrangian cost ($J_{min}(\lambda)$), obtained by the summation of the block Lagrangian costs, will also be minimized.

5.3.3 Dynamic Programming for coefficient selection

For the *coefficient selection* approach we need to select the coefficients from a single block such that the block Lagrangian cost will be minimized. Finding the optimal solution is a linear programming problem. This can be solved using a (Viterbi like) dynamic programming algorithm. This algorithm was originally presented by Ramchandran and Vetterli [Ramc94] for the rate-distortion optimal zero thresholding, i.e. adaptive quantizer zero-zone sizing, in MPEG encoders. This thresholding is equivalent to our problem of coefficient selection and the algorithm is therefore adapted to our problem. We will briefly discuss the dynamic programming algorithm.

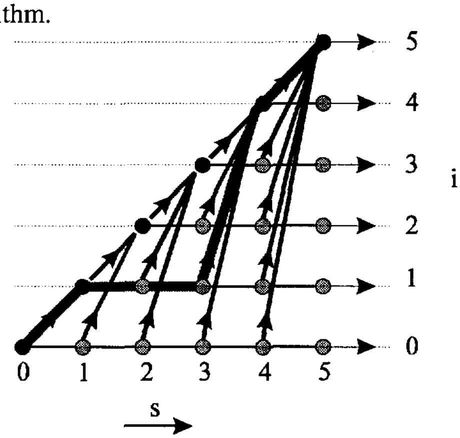


Figure 5.5: Lagrangian cost structure (C=5)

Given a DCT block of the input picture with C non-zero coefficients in the coefficient domain, a cost structure as shown in Figure 5.5 can be constructed (C=5). In this figure the nodes on the C+1 horizontal propagation line store the minimal Lagrangian cost $J_i(\lambda)$ associated with including coefficient i as the last coefficient, where $i \leq C$ and i=0 corresponds with *no* coefficients. The figure's horizontal dimension (s) represents the consecutive steps in calculating a certain $J_i(\lambda)$; at the step s=i the minimal cost to end at coefficient i is calculated. Only the black nodes are processing units, the other nodes are used to horizontally propagate the minimal Lagrangian cost to end with a particular coefficient. For i=0 the cost of including no coefficients is given by:

$$J_0(\lambda) = \sum_{0<k\leq C} E_k^2 \qquad (5\text{-}6)$$

where E_k represents the de-quantized contribution of the coefficient k. The diagonal branches store the change in Lagrangian cost when retaining coefficient i following coefficient k. This cost is calculated as follows:

$$\Delta J_{k,i}(\lambda) = -E_i^2 + \lambda \cdot R_{k,i}\ ,\ 0 \le k < i \le C \qquad (5\text{-}7)$$

where $R_{k,i}$ is the bit cost of including coefficient i after coefficient k. $R_{k,i}$ is determined by look up in the MPEG VLC tables, using the zigzag distance between the two coefficients as the run length.

The optimization now goes as follows. For each coefficient $0<i\le C$ the minimal Lagrangian cost $J_i(\lambda)$ is calculated by taking the minimum of the sum of the cost of all possible predecessor coefficients and the incremental cost of adding the current coefficient as follows:

$$J_i(\lambda) = \min_{0\le k<i}[J_k(\lambda) + \Delta J_{k,i}(\lambda)] \qquad (5\text{-}8)$$

For each coefficient the minimal Lagrangian cost and the corresponding optimal predecessor is retained and propagated horizontally. Therefore, at step s=i, we know for coefficient i what its minimal Lagrangian cost is and what the required predecessor is to achieve this cost. By back tracing the optimal predecessors the path of used coefficients that corresponds with the minimal Lagrangian cost to end with coefficient i, can be found.

After C+1 calculation steps, when the cost of ending in each of the coefficients is calculated, the minimal block Lagrangian cost and the optimal last coefficient are obtained by selecting the coefficient with the minimal cost:

$$J_b(\lambda) = \min_{0\le i\le C}[J_i(\lambda)] \qquad (5\text{-}9)$$

The optimal coefficient selection is found by back tracing the optimal predecessors from the optimal last coefficient.

5.3.4 Minimization for codeword selection

To obtain the optimal *codeword selection* in the codeword selection approach an equivalent procedure can be used. The codeword selection is equivalent to the coefficient selection with limited freedom on the allowed coefficients to be selected; from each block a number of c coefficients can be used, starting from the first coefficient and ending with coefficient c, where the cut-off level $c\le C$. The problem can be formulated as a simplified coefficient selection. In this simplified formulation only the branches between the immediate neighbor coefficients of Figure 5.5 will be valid, i.e. Equation (5-7) is only valid for $k = i-1$ and referring to Equation (5-8) the Lagrangian cost per coefficient is obtained as follows:

$$J_i(\lambda) = J_{i-1}(\lambda) + \Delta J_{i-1,i}(\lambda)\ \ \forall 0 < i \qquad (5\text{-}10)$$

When we end with a certain coefficient in the codeword selection approach, all the predecessor coefficients will be used. The calculation of the Lagrangian cost per codeword becomes a simple addition of the Lagrangian cost of the previous coefficient with the incremental cost of adding the current coefficient. After C+1 calculation steps the minimal Lagrangian block cost is obtained by selecting the coefficient with the minimal cost as the last coefficient and using all coefficients up to and including this last coefficient (Equation(5-9)). As such, no dynamic programming algorithm will be necessary to find the optimal solution.

5.3.5 Evaluation of Coefficient and Codeword Selection Method

We experimentally evaluate both the optimal *coefficient selection* (O-CF) and the optimal *codeword selection* (O-CW) solutions, corresponding with Figures 5.3.b and 5.3.c respectively. The bit rate of the normal play sequence is 9.6Mb/s. A large section of the R-D curve will be plotted by performing a sweep with the Lagrange multiplier λ; for each λ the R-D operation point with the minimal Lagrangian cost is plotted. The distortion measure used in the plot is the weighted mean squared error (WMSE) which is determined as follows:

$$WMSE = \frac{1}{\#\,blocks}\sum_{b=0}^{\#blocks}\sum_{q=1}^{C} E_{b,q}^2 \qquad (5\text{-}11)$$

where for each block b, $E_{b,q}=E_q$ for the not selected coefficients and $E_{b,q}=0$ for the selected coefficients.

The lower curve of Figure 5.6 (O-CF) shows the resulting R-D curve when performing the optimal *coefficient selection* on a particular input frame. This curve represents the constraint on all sub-optimal coefficient selection solutions that use an algorithm with a reduced complexity.

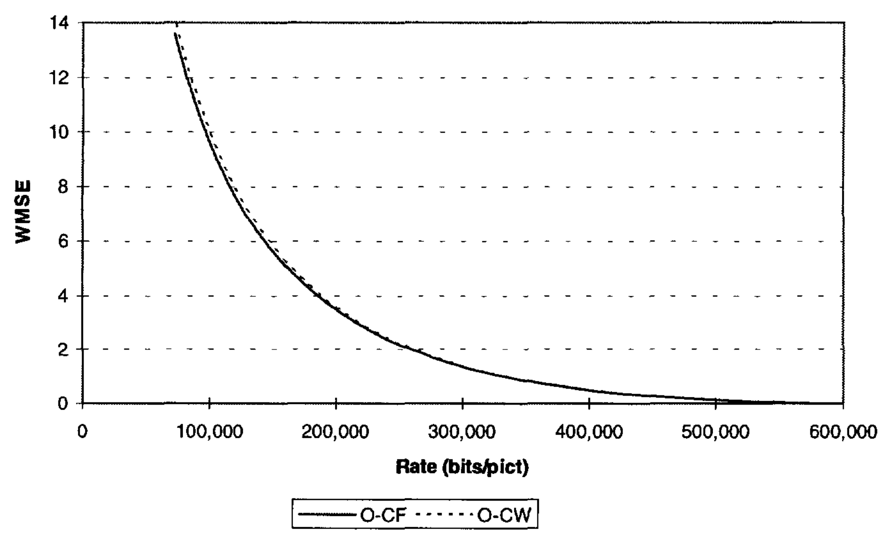


Figure 5.6: R-D curve for optimal (a) and constrained optimal (b) solution for an entire frame.

The upper curve of Figure 5.6 (O-CW) shows the R-D curve for the optimal *codeword selection* for the same input frame. From the figure it can be seen that very little difference exists between the R-D curve of the optimal extraction of either coefficients or codewords. Similar results were obtained for different frames. When the same experiment was performed for normal play sequences of lower bit rates then even less difference exists between codeword and coefficient selection. We can thus conclude that little advantage is gained from the total freedom in the coefficient selection and a limited selection of some codewords per block is sufficient. As the computational complexity of the selection algorithm (per block) of the optimal coefficient approach is larger than that of the optimal codeword approach, it is clear that the optimal codeword solution is computationally far more attractive.

To compare these *optimal* results to the *zonal based codeword selection* (Z-CW) approach, an equivalent R-D plot must be found for the zonal approach. However, the zonal approach only has a limited number of R-D operation points. Because a constant cut-off level is selected for the entire picture only N^2 different cut-off levels are possible and therefore only N^2 rate-distortion combinations are possible.

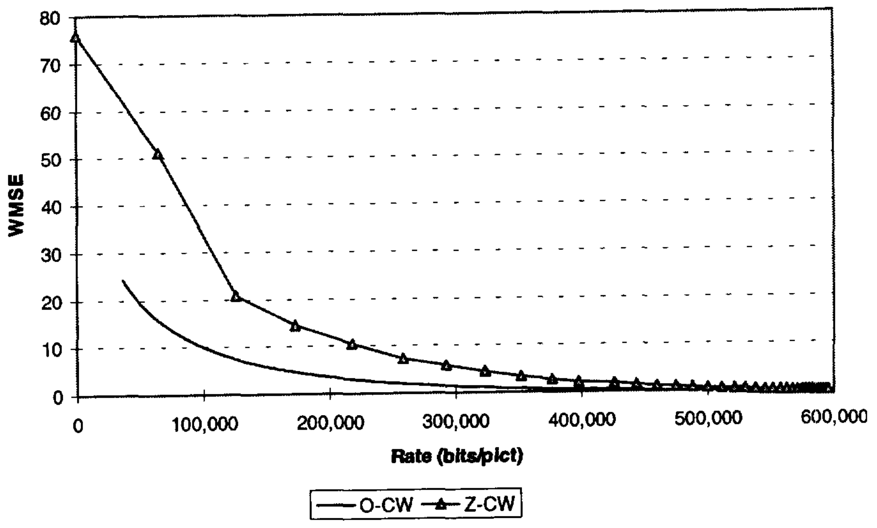


Figure 5.7: R-D curve of zonal and optimal codeword approach for an entire frame.

In order to evaluate this zonal approach, irrespective of the particular rate control mechanism, the R-D curve for a particular frame is plotted in Figure 5.7 (Z-CW), where the operation points (triangles) are achieved by sweeping the cut-off level from 0 to 63 for the entire picture. In an actual system, which includes a rate control mechanism, the operation point will be somewhere between two calculated

operation points, where the exact position is determined by both the data and the rate control mechanism. Therefore the line connecting the operation points in Figure 5.7 merely suggests what the R-D curve may be like in practice.
As a comparison, Figure 5.7 includes the R-D plot of the *optimal codeword solution* (O-CW) taken from Figure 5.6. It can be seen that the zonal approach is inferior to the optimal codeword selection; i.e. the optimal solution achieves a large gain by selecting a cut-off level for each block separately. In particular this is the case in the range of 40,000 to 200,000 bits per picture, which is the important range for the targeted bit rate around 1Mb/s.
The complexity of the zonal approach is lower than that of the optimal codeword selection approach, in particular with respect to the required rate control mechanism. In the next section the rate control mechanism will be evaluated.

5.4 Bit stream Extraction Rate Control

5.4.1 Optimal Forward Rate Control

Using the selection mechanisms described in the previous section, it is now possible to extract a low bit rate dedicated fast playback stream from the input MPEG encoded bit stream by selecting appropriate intra coded frames and setting a rate budget for each consecutive frame. The free parameters are the frame rate multiple (k_m) and the target bit rate of the resulting fast playback sequence; for a specific I frame these two will set the picture bit budget R_{budget}.
The desired optimal slope λ_o that satisfies Equation (5-3) is not known a priori. It can be obtained using a convex search for the following maximization which has a guaranteed unique maximum due to the convexity of the problem [Shoh88]:

$$\lambda_o \leftarrow \max_{\lambda \geq 0}[J_{min}(\lambda) - \lambda \cdot R_{budget}] \qquad (5\text{-}12)$$

The approach to finding either the optimal coefficient extraction (O-CF) or the optimal codeword extraction (O-CW) to the constrained problem (5-1) leads to an architecture as shown in Figure 5.8 for each picture. For a certain value of λ the coefficient/codeword selection can be performed for each block separately (block extraction) by minimizing the block Lagrangian cost $J_b(\lambda)$. Given the optimal solution for the entire picture, the rate and distortion for the particular λ, are determined. Using the iterative search, the optimal slope λ_0 that satisfies (5-3) can be approximated; the optimization is repeated as λ converges to λ_0.

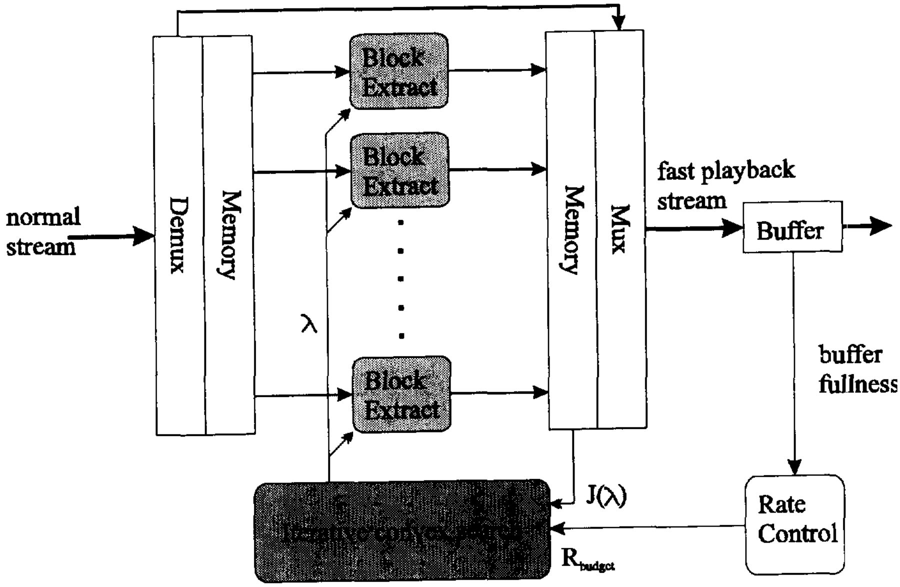


Figure 5.8: Optimal forward rate control architecture

The most important hardware requirement is the storage of the parsed VLC codewords or the coefficients of an entire frame such that the iteration can take place to find the optimal coefficient/codeword selection for the frame. This is a very large memory requirement which implies that a whole frame must be buffered to use the approach in Figure 5.8.

5.4.2 Sub-optimal Extraction

5.4.2.1 Lagrangian Feedback

With respect to the rate control within each individual picture the architecture of Figure 5.8 can be interpreted as a feedforward rate control. In this forward rate control advantage is gained from the fact that the contribution of each codeword on both the rate and the distortion is known and the optimal codewords to satisfy the rate constraint per frame can be selected.

By replacing the iterative feedforward rate control of the O-CW approach by a feedback approach, the significant memory requirement can be alleviated. Such a feedback rate control, is shown in Figure 5.9. The λ control parameter will now be approached iteratively for the consecutive blocks, performing one iteration per block. Essentially, the iterations operating on an entire frame are transformed into a recursion over the macroblocks within a frame.

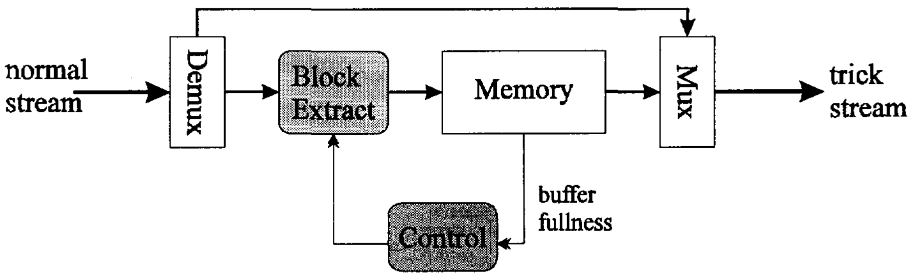


Figure 5.9: Feedback rate control

At the stable state, the control parameter λ will be kept as constant as possible such that the optimal solution (O-CW) of a constant slope λ on the R-D optimal curve is approached. To achieve this nearly constant control, the bit buffer feedback mechanism is required to be slow enough to be insusceptable to the noise in the buffer fullness measure. At the same time the rate constraint can be satisfied. The great advantage of this *Lagrangian feedback codeword extraction* (LF-CW) solution is that on the fly extraction without buffering an entire picture is possible.

5.4.2.2 Simplified Lagrangian Feedback Extraction

The above method can be further simplified to reduce its complexity. To this end we assume that the encoder used a relatively constant quantization step for the whole picture. As a consequence the term E_k of Equations (5-6) and (5-7) does not need to represent the de-quantized contribution and instead it simply represents the quantized value of the coefficient. The significance of this approach is that there is no need for de-quantization to carry out the coefficient selection, i.e. the de-quantizer (DQ) of Figure 5.3.c will not be used. Consequently it will not be necessary to interpret the MPEG syntax with respect to the dequantization scale and mode. A possible drawback is that the performance becomes dependent upon the variance in the quantizer scale used by the original encoder. The *simplified Lagrangian feedback codeword extraction* (SLF-CW) will be compared to the other approaches in the system's overall evaluation.

5.4.2.3 Zonal Based Extraction with Feedback Control

The *zonal based codeword extraction* mechanism needs to be subjected to a rate control in a similar manner as the Lagrangian feedback mechanism. The architecture is therefore similar to Figure 5.9. As no tuple decoding is performed for this method the only remaining control parameter is the cut-off level. The cut-off level for a particular block is now governed by a feedback rate control algorithm which monitors the buffer fullness. The cut-off level will be controlled such that the rate of the picture remains as close to the target as possible. In order to keep the image quality as constant as possible, the cut-off level will only be allowed to vary slowly. The resulting zonal feedback codeword selection (ZF-CW) is a further simplification of the SLF-CW method.

5.4.3 Fast Playback Extraction Method Summary

#	Name	Full Name	Fig 5.3	Rate control	Method	Complexity	
						per block	rate control
1	TRANS	Full transcoding	a	Usually feedback	new quantization	high	low
2	O-CF	Optimal coefficient	b	Optimal feedforward	de-quantize, select coefficients	medium	high
3	O-CW	Optimal Codeword	c	Optimal feedforward	de-quantize, select codewords per block	low	high
4	LF-CW	Lagrangian feedback codeword	c	Buffer feedback	select cut-off level based on de-quantized contribution	low	medium
5	SLF-CW	Simplified Lagrangian feedback codeword	c	Buffer feedback	select cut-off level based on quantized contribution	low	low
6	ZF-CW	Zonal Feedback codeword	c	Buffer feedback	link cut-off level to buffer fullness	very low	low

Table 5.1: Summary of extraction schemes and their properties

Table 5.1 summarizes the discussed fast playback stream extraction methods. This table may be used as a reference in the following evaluation.

5.5 Evaluation

A 20 frame fast playback stream was extracted from a 240 frame 9.6 Mb/s Albertville Olympics sequence (speedup n=3.0, frame rate fraction k_m=4, fast playback bit rate R_{TRICK}=1.25Mb/s) using the described extraction methods: O-CF, O-CW, LF-CW, ZF-CW. The SNR between the decoded fast playback pictures and the corresponding pictures of the *original sequence* was measured and the results are plotted in Figure 5.10. The following SNR definition was used:

$$SNR = 10\log_{10}\frac{\sigma^2}{MSE} \text{ (dB)} \tag{5-13}$$

where σ^2 is the variance of the original image and MSE is the mean squared error of the coded image. From the figure it can be seen that for any particular extraction method a large variation in the SNR exists when considering consecutive pictures. The reason for this is that, as only one out of every 12 frames of the normal play stream is used, a scene in the fast playback signal often consists of no more than 3 pictures.

We note that the commonly used ZF-CW on the average performs 3dB worse than the other extraction methods. We also notice that there is almost no difference between the optimal coefficient and optimal codeword (O-CF and O-CW) approach such that we can confirm that for our specific case of fast playback stream extraction there is little advantage to this complex method.

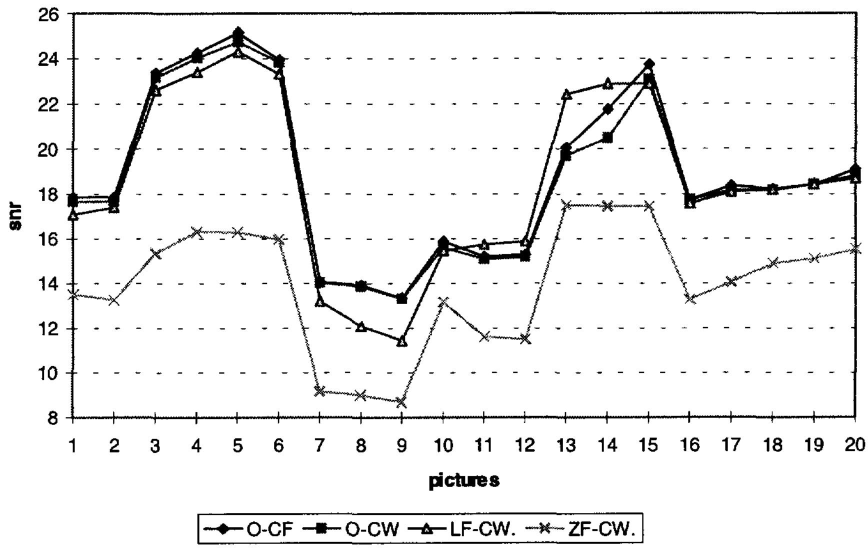


Figure 5.10: SNR plot of typical extracted sequence (n=3.0, T_m=4, 1.25Mb/s).

The LF-CW method, which tries to approach the O-CW using a low complexity feedback rate control instead of the full forward rate control, manages to stay very close to the optimum. For some pictures the difference in SNR is negligible, for others it can become significant. Note that these results are largely dependent upon the particular feedback mechanism.

For certain frames LF-CW is better than O-CW. This is merely a coincidental phenomena caused by the slowly adapting feedback mechanism. It does not reach the target rate for every frame separately. Bits that were not used for one frame are available for the next and the quality of that particular frame will thus be better than the limited optimum which satisfies the bit budget.

Figure 5.11 presents two additional experiments: the full transcoding solution (TRANS) and the simplified Lagrangian feedback codeword selection (SLF-CW). The figure shows that, as expected, transcoding yields a better result than the codeword extraction methods. On the average the transcoding solution is about 2dB better than LF-CW, which in turn is about 4dB better than ZF-CW. The SLF-CW mechanism is only slightly worse than the LF-CW mechanism. Given its very low complexity, the SLF-CW is a promising real-time fast playback stream extraction method.
Based on visual observations we can confirm that the improvement from the zonal feedback method to the Lagrangian feedback method is significant; for the chosen bit rate and frame rate combination most of the blocking artifacts have disappeared. The visual improvement when using the full re-coding method is also apparent but less significant.

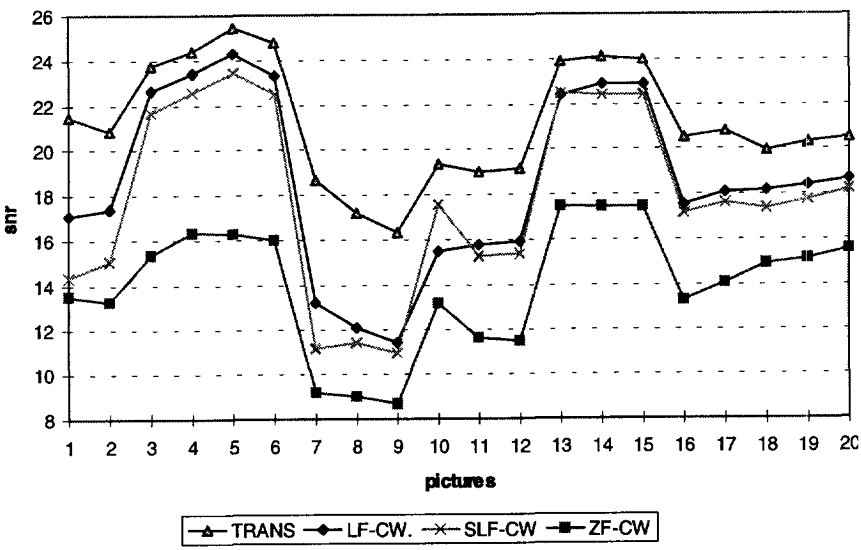


Figure 5.11: SNR plot of extracted sequence using feedback rate control mechanisms

6. Hardware Fast Playback Verification

6.1 Verification Hardware

6.1.1 Objective

The objective of the verification hardware is to implement and verify a basic hardware adaptation interface for MPEG recording on a DART recorder. Using this adaptation interface the model based tape formats developed in Chapter 3 can be verified. The formatting (mapping) will be performed in real-time, using a down loadable mapping table. Both the rudimentary fast playback and dedicated stream formatting will be evaluated with the verification hardware.
The studied application is the recording of standard resolution MPEG-2 video with a maximum bit rate of 10Mb/s. To this end the bit machine with the usable bit rate of 12.5 Mb/s (M4) should be used. The adaptation interfaces were, however, defined flexibly to allow different recorders (M2: 25Mb/s and M1: 50Mb/s) to be used in the evaluation; for our particular experiments the behavior of an M4 machine was simulated on an actual M2 machine.

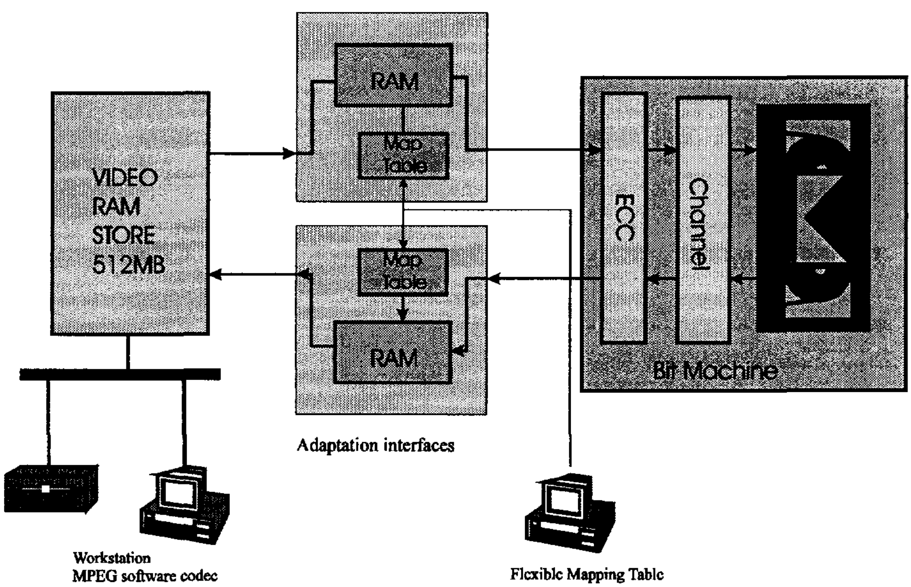


Figure 6.1: Verification hardware concept

6.1.2 Concept

The concept of the DART verification hardware is shown in Figure 6.1. In the verification hardware both the encoding and the decoding are done off-line in software on a workstation. The adaptation layer which involves the packetizing of the data in sync-blocks and the mapping of the sync-blocks onto the tape, is the first element of the real-time hardware chain. It provides the link between the mechadeck, the channel electronics and the error correction unit on one side and the Video RAM store system which is the interface to the computer system on the other side. The usage of this RAM buffer is necessary to be able to achieve a high enough i/o bit rate to the computer system. Separate adaptation interfaces exist for recording and playback.

Figure 6.2 shows a photo of the setup used for the verification hardware. At the right of the picture the bit machine developed by Philips Research can be seen with (from right to left) the experimental mechadeck, the channel rack and the ECC rack. The adaptation interface rack developed by the Information Theory Group of the TU Delft forms the interface from the hardware chain to the computer environment.

Figure 6.2: Verification hardware setup (photo courtesy of Philips Research, taken from [DART12])

6.1.3 Adaptation Interface Flexibility

The adaptation interface is the dedicated MPEG interfacing provision of the DART:

- It packetizes the data in sync-block sized packets.
- It supports the usage of various experimental bit machines.

- It performs the adaptation from the bit rate of the service to be recorded to the operating bit rate of the verification hardware, i.e. it adds the appropriate amount of dummy sync-blocks to the bit stream.
- It performs a prescribed formatting of the sync-blocks to be written on the tape. This formatting, i.e. the re-mapping of the data, is required by both the contiguous format of rudimentary fast playback and for the formatting of dedicated trick streams.
- It provides an interface to the outside world that is independent from the way data is put on tape. The bit stream is entered directly to the recording interface and it is played back directly from the playback interface. The nature of the fast playback burst will be changed at the interface point. As such, a virtual recorder is created, which may have different characteristics than those of the original recorder.

All these roles will be implemented in a single mapping step which is described by an addressing table to be used by both the recording and the playback interface. In the design of the single mapping table each of the above roles can be considered separately and the resulting map can be formed by a concatenation of the separate maps that perform the separate operations.
The de-mapping interface performs the inverse operation of the mapping interface and de-shuffling the data to the appropriate output sequence.

6.1.4 Hardware Architecture

A brief description of the adaptation interface hardware will help to understand the possibilities of the system. Both the recording and the playback adaptation interfaces consist of a Dual Data Frame Memory (DFM), a down loadable addressing table and some control logic.

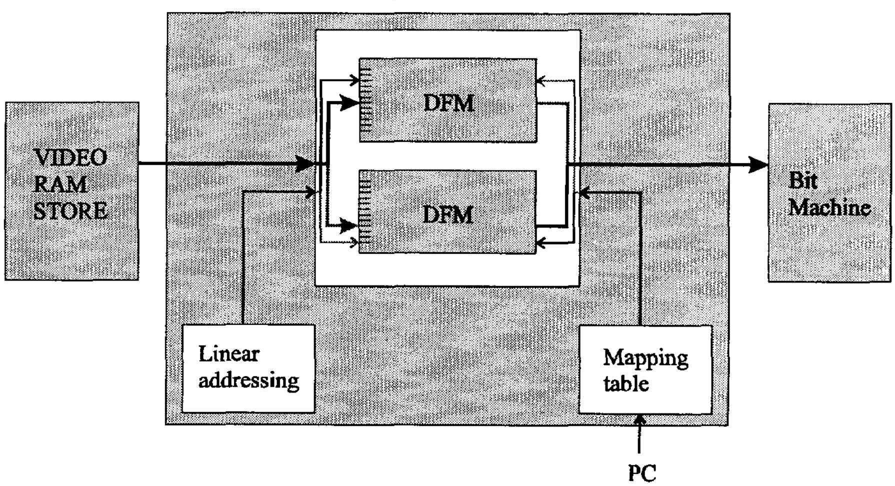


Figure 6.3: Recording interface architecture

In the recording adaptation interface (Figure 6.3) the DFM memories are used alternately for inputting data from the outside world and for transmitting sync-blocks to the recorder (ping-pong memory). Each DFM can contain a Data Frame Structure (DFS) or equivalently up to 12 tracks of packetized MPEG information. The organization of the memory is sync-block based, i.e. the sync-block numbering is part of the address of the memory bytes. At the input the DFM is filled with a DFS of data from the source (DVS RAM) using a linear addressing. The sync-block section of the addressing of the output of the DFM is governed by the down loadable mapping table. As such, the sync-blocks stored in the DFM are sent to the recorder in an order prescribed by the mapping table. Implicitly this allows the sync-blocks to be placed at exactly prescribed locations on tape.
The playback adaptation interface has an inverse operation with an equivalent architecture. Sync-blocks that are input from the tape are preceded by a header which contains the sync-block number as stored on tape. The sync-block number is used as address of the down loadable mapping table to find the DFM location at which the particular sync-block should be stored. Two different down loadable mapping tables exist on the interface, one for use in normal playback and one for use during fast playback. The sync-blocks that belong to a single DFS are collected. When the boundary of the DFS data on tape is encountered then the DFM operation toggles such that data of the new DFS is collected in the other DFM. The collected data is output as a single depacketized burst to the DVS RAM system. Any missing sync-blocks are replaced by an inserted sync-block that contains the dedicated MPEG error start code. This signals the detected error to the decoder.

6.2 Fast playback Model Verification

6.2.1 Overall Observation

Using the verification setup different runs of writing data on the tape and performing a fast playback of n=3 were performed. Each time 40Mbyte were read back to the computer system.
The model for the fast playback signal developed in Chapter 3 predicts that for a speedup factor n, where n is a fraction of the amount of tracks in a frame (e.g. n=3.0), the same data sections will be read from consecutive frames, i.e. a single line from the phase-read diagram will represent all the sections read from the consecutive frames. Since there is no phase lock between the scanner and the tape an uncertainty of the actual phase exists, but the phase has been assumed to be constant for the entire fast playback session.
The actual recording system used in the verification hardware had no frequency lock between the scanner and the tape during fast playback. This means that the speedup factor n is not the exact integer but a small error exists in the speedup factor ($n=3.0 \pm \delta$). For the purpose of this discussion we assume that there is a frequency lock and the phase is slowly varying across consecutive tracks. This means that

$$p_{i+1}=p_i+\phi, \tag{6-1}$$

where p_i is the phase for DFS i and ϕ is the small phase difference between consecutive tracks.

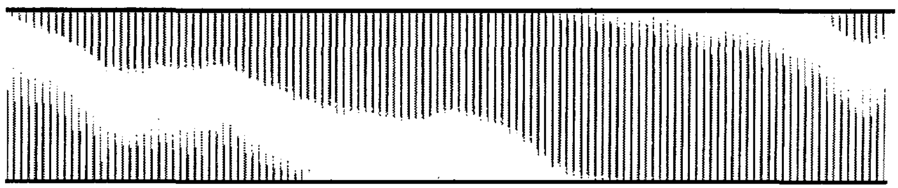


Figure 6.4: Recovered sections from tape for the M2 machine (n≈3)

In Figure 6.4 a tape section is shown with the tracks drawn perpendicular to the tape travel direction. Only the recovered sections of the tracks are drawn. The figure is typical for the sections recovered from tape with the verification hardware at a fast forward speedup factor of n≈3, using a plain mapping table.
Figure 6.5 is a transformed version of Figure 6.4, where on each horizontal line tracks from a single DFS are plotted head to tail and consecutive DFSes are plotted vertically on top of each other. The shaded areas correspond with the recovered data sections. Two hundred consecutive DFSes are plotted, corresponding to 5.3 sec. of recovered fast playback data.

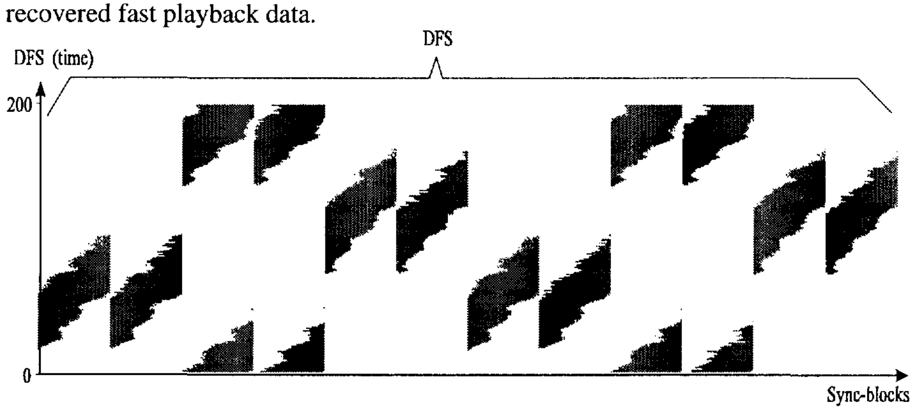


Figure 6.5: Recovered sections from tape for the M2 machine (n≈3.0)

A comparison of Figure 6.5 with the phase-read diagrams of Chapter 3 seems obvious at this point. It is important to note the difference between these two types of figures. The phase-read diagrams of Chapter 3 are the result of a model, where it is possible to plot the read results for a single DFS for any possible phase. In Figure 6.5 the vertical axis represents consecutive DFS's read from tape. By a coincidence the scanner to tape phase at the beginning of the consecutive DFS's has a slight drift, such that the DFS's have a different phase and the resulting figure has many commonalties with the phase-read diagrams.

By analysis of the periodicity of the recovered pattern we can deduce that after about 200 DFS's a phase shift of 3 double tracks has occurred. This means that the error in the tape speed is about 0.25%. Under the assumption of a frequency lock and a shifting phase this is equivalent to a shift of

$$\phi=3*10^{-2} \tag{6-2}$$

per frame, where the track pitch is the unit phase.

It is important to note that for our experiments this phase drift per DFS forms a distinct advantage. It provides the opportunity to analyze the read mask for every possible scanner to tape phase in a single experiment. In an indirect manner, a sweep of the entire phase-read diagram is thus performed. Collecting the same spread of data with a speed locked bit machine would involve a very large amount of experiments.

6.2.2 Estimation of Q-factor

We recall that in the mapping design for dedicated stream formatting, the fraction read (f_{READ}) per DFS is an important factor. Evaluation of the fraction read will give an indication of the Q-factor of the system.

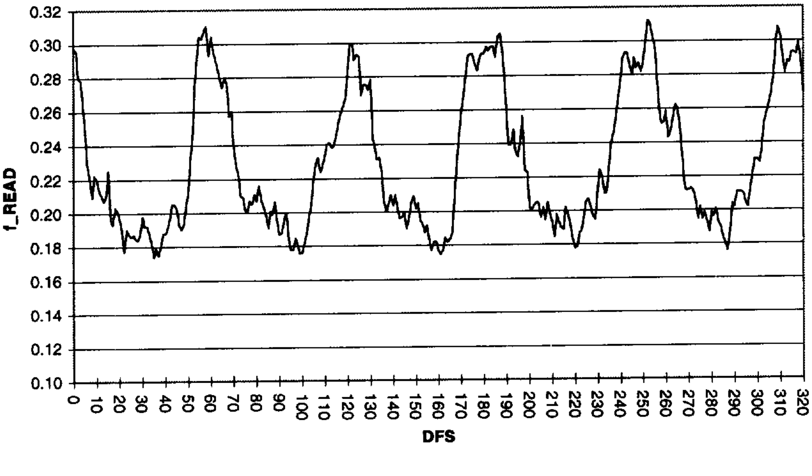


Figure 6.6: f_{READ} for consecutive DFS for n≈3

A typical plot of f_{READ} as a function of the frame number (time) is given in Figure 6.6. The fraction of the data read per DFS periodically cycles between about 0.18 and 0.3 . Based on the fast playback model of Chapter 3 we would expect f_{READ} to be more constant; irrespective of the phase about the same amount of data should be read from a frame. The cause of the discrepancy between the model prediction and the observed values is the fact that the model assumes a 180° wrapped track. In the actual system the video section of the track format is 138 sync-blocks large, while it would take 186 sync-blocks for the full 180°. Consequently, the video section is wrapped over 133.5°. The remaining sync-blocks are used for the audio block, vertical error correction codes and some editing gaps.

The periodicity of Figure 6.6 can now be understood. The phase of the scanner to the tape determines what parts of the entire track (180°) are recovered. For certain phases more of the non-video areas are recovered, for other phases only video sync-blocks are recovered. For a system that has a frequency lock during fast playback, f_{READ} would be constant and dependent upon the phase between the scanner and the tape. The actually used system without a frequency lock cycles through the different phases by having a phase offset from one frame to the next. While the fraction read from the entire tape (i.e. the sum of the video and the non-video areas) is constant, f_{READ} for the video area will have a periodic nature.

A recording system without any phase lock has to take into account the worst case for f_{READ} which is equal to the minimum of the system without a frequency lock. The lack of a frequency lock will therefore not be a direct disadvantage for the data recovering capabilities of the free phase system.

The actual Q-factor can be estimated from the maximum f_{READ}; at the maximum the entire data section read lies in the video area of the tape format. Given that for the video section $f_{READ}=0.3$, we can derive f_{READ} of the entire track to be $f_{READ}=0.3*(133.5/180)=0.22$. With E=1 and n=3.0 we conclude that Q=0.67 .

Working with the worst case and considering only the video section the Q-factor is estimated from the minimum of f_{READ} to be: Q=0.54 . It is this factor that should be used in the design of the different mapping tables. In Chapter 3 we used Q=0.5 for the design of most tape formats. At this point, this assumption for the Q factor is confirmed to have been a good choice.

6.2.3 Model Correction

Based on the observations of the tape sections read during fast playback, it can be verified that the model adequately predicts the read mask. Only small differences exist between the prediction and the actual read mask.

The immediately noticeable difference between the model and the actually read mask is the previously discussed discrepancy between the expected and actual speedup factor. This means that the slope of the head trace across the track is slightly different from expected. As this difference is a statistical error, it is not useful to correct for it by adapting the model. We assume that on average, measured over different systems, the speedup will be as expected.

The most important deviation of the model from the actual system originates from the fact that the model does not account for the exact geometrics of the system. Instead it is based on the abstraction of tracks that are written perpendicular to the tape travel direction and it fails to take into account the exact positioning of the individual heads of the head pair on the head wheel. The model thus predicts that if a sync-block of one track of the track pair is read, the corresponding sync-block of

the other track of the pair will be read as well. This is, however, not necessarily the case in practice.
The exact modeling of the difference between the track sections read from the individual tracks of a track pair requires full knowledge of all geometric parameters. Alternatively, the distance between the heads of a head pair of the model can be adjusted to correspond more with the observed read mask. By comparing masks from the adjusted model with the actually read mask, minimizing the difference between the two, we can conclude that using a inter head distance of $1.1*p_t$ corresponds more with the observed read pattern. This adjusted model will be used in the mappings to be used in the following sections.

6.3 Rudimentary Fast Playback

6.3.1 Contiguous Format

The contiguous format is expected to yield better rudimentary fast playback results due to the fact that many short bursts are grouped into a single long burst within a DFS. In Chapter 4 this was extensively evaluated by simulation. We will perform the same evaluation using the actual system.
Figure 6.7 shows the resulting burst of data read when the contiguous map is used by both the recording and playback adaptation interface. Essentially this figure is the contiguously re-mapped version of Figure 6.5. The figure shows that the contiguous re-mapping, grouping the individual bursts from a DFS, works as expected, leaving a single burst per DFS of which the length is a function of the speed-up and the Q factor (f_{READ}).

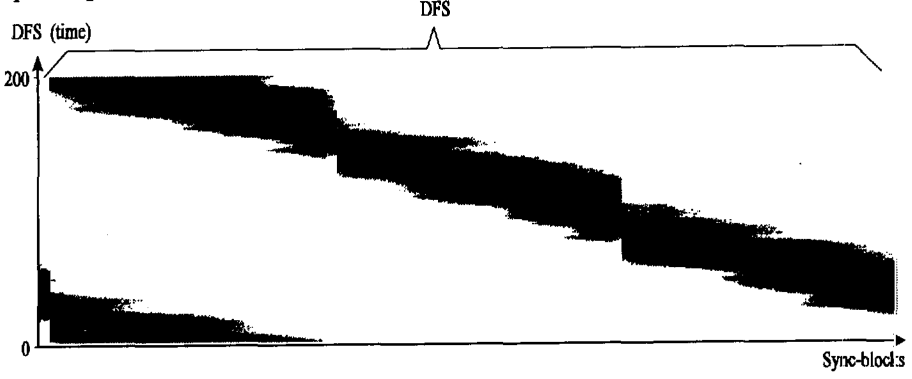


Figure 6.7: Contiguously re-mapped DFS sections for the M2 machine.

In Chapter 3 the concept of enhanced fast playback (E=2) for the M4 machine was introduced, where the largest amount of data possible is recovered from tape by leaving the head pair switched on all of the time. Figure 6.8 shows the result of the contiguous re-mapping with enhancement for the M4 machine. When compared to Figure 6.7 f_{READ} is doubled.

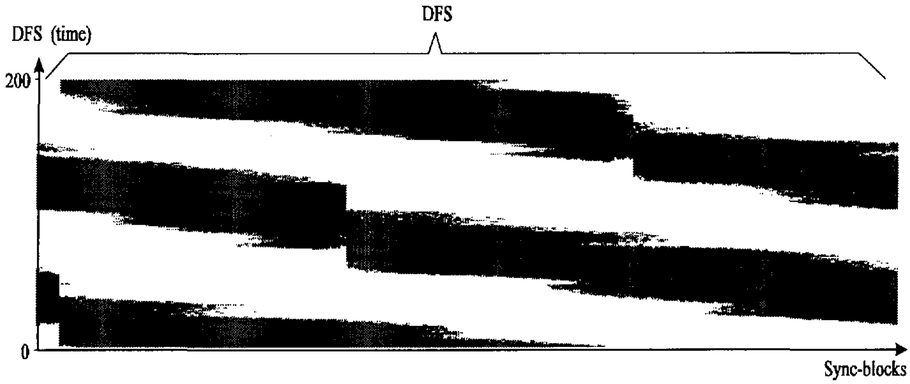


Figure 6.8: Contiguously re-mapped DFS sections with enhancement (E=2) for M4 machine

6.3.2 Effect of Model Failure on Contiguous Burst

Despite the attention paid to the model and its adaptation to the observed data, it does not exactly predict the sync-block pattern read. In particular, the speedup error (±0.25%) is not compensated for. It is thus important to analyze what effect this model deviation has on the nature of the fast playback bursts for the contiguously re-mapped data bursts.

During fast playback a number of bursts of the DFS are read. The contiguous mapping is designed in such a way that the centers of the individual bursts from tape are mapped to the center for the contiguous burst, the edges of the individual bursts will be mapped to the edge of the contiguous burst. The model is accurate enough to predict the middle of the individual data bursts that are read. However, at the edge of the individual bursts errors may occur.

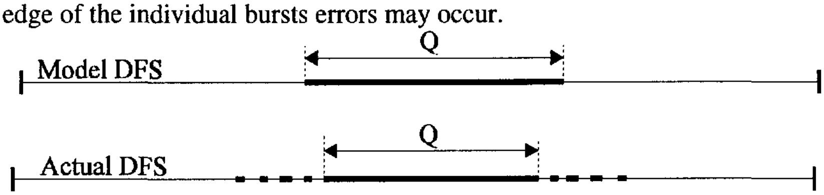


Figure 6.9: Contiguous burst for model and for actual recorder

The error in predicting the edge of the bursts read from tape results in a breakdown of the contiguous re-mapped burst at the edge. Figure 6.9 shows the difference between a burst in a DFS for the model and a typical re-mapped actual burst. The breakdown at the edge of the burst can also be seen in Figure 6.7.

Due to the model failure at the edge of the burst, the contiguous burst of the actual system has the same length as an ideal burst with a lower Q-factor. For the further mapping design, a smaller Q-factor should therefore be used to compensate for the model deviation.

6.3.3 Evaluation

To evaluate the performance of the rudimentary fast playback, experiments were performed with several encoded versions of a the "123" Albertville Olympics sequence. All versions were encoded at 9.6Mb/s such that effectively 1247 sync-blocks of the 1656 sync-blocks per DFS were required. The *dummy* spare sync-blocks (~25%) actually stored *real* data; the first sync-blocks were repeated in the dummy area. This usage of the spare capacity, which is simply done by the appropriate mapping design, will slightly enhance the fast playback signal.
Two different GOP structures were used to study the effect of the GOP length on the resulting signal; both a GOP length of 13 frames (GOP13 = IBBPBBPBBPBBP) and a GOP length of 3 frames (GOP3 = IBP) were used. Furthermore, for both GOP structures, three slice structures were used: 3 macroblocks per slice (3mb), 15 macroblocks per slice (15mb) and 45 macroblocks per slice (45mb). For the 15mb version with both GOP3 and GOP13 a fast playback signal of 600 frames was recovered for the speed of n=3.0±0.25% For the other slice structures a fast playback signal of 300 frames was recovered. Table 6.1 gives the experimentally determined average picture age and the corresponding standard deviation (over 300 trick pictures) for the two GOP structures and different slice lengths. Both the plain (pl) and contiguous (ct) formats were used.

	GOP3	*(pl)*	*GOP3*	*(ct)*	*GOP13*	*(pl)*	*GOP13*	*(ct)*
	average	*sdev*	*average*	*sdev*	*average*	*sdev*	*average*	*sdev*
3mb	3.10	1.73	1.41	0.55	8.79	2.60	7.18	2.20
15mb	3.45	1.79	1.58	0.67	8.95	2.40	7.45	2.14
45mb	3.91	1.87	1.73	0.75	10.26	2.89	7.92	2.26

Table 6.1: Picture age (n≈3)

The nominal speed of n=3.0 is a taboo speed for rudimentary fast playback. However, due to the use of a system without a frequency lock between the scanner and the tape the predicted reading periodicity with respect to the picture sections is luckily less significant in practice. Nevertheless, by comparing the results of Table 6.1 for the short GOP lengths with those of Chapter 4 (Table 4.1), we see that the average age is higher than expected for the non-taboo speeds. This is particularly the case for the plain format. Visually, in the video sequence, this hovering around the taboo speed does at certain moments become apparent by a low refresh rate for a certain picture section. We can thus confirm experimentally that, in an actual system, care needs to be taken to stay clear of the taboo speeds.

From the table it can be seen that using long slices will cause the average macroblock age to increase; more time is needed to re-synchronize when starting to decode data bursts. This effect is particularly strong for the plain format where for every short data burst a section will be lost during re-synchronization. For the longer bursts of the contiguous format less data is lost during re-synchronization and the average macroblock age thus becomes smaller.

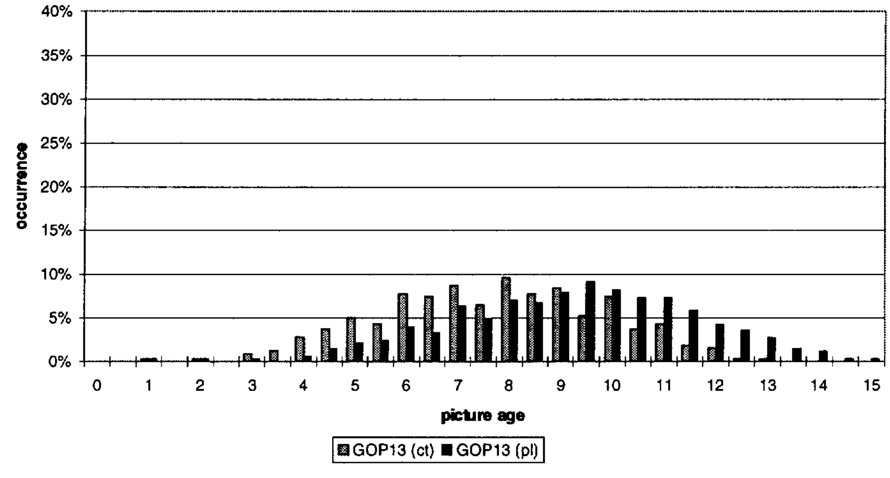


Figure 6.10: Picture age histogram for GOP13 (15mb, n≈3)

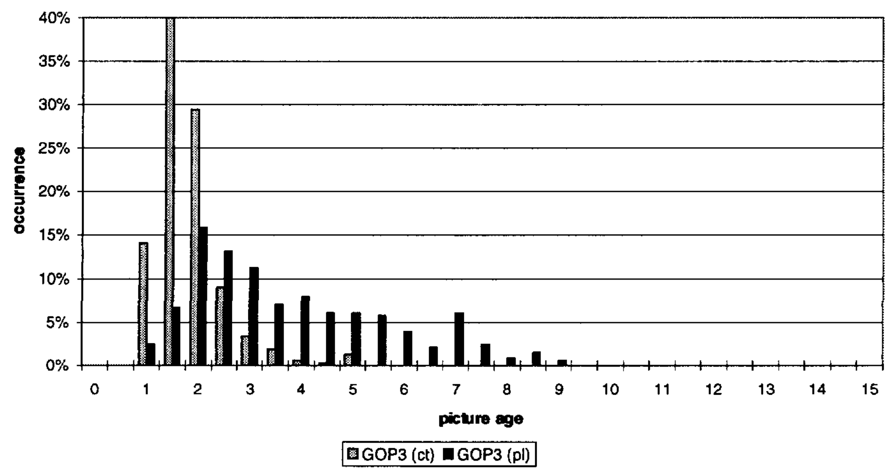


Figure 6.11: Picture age histogram for GOP3 (15mb, n≈3)

The histograms of the picture age, corresponding with the 15mb version of the above measurements, are given in Figure 6.10 and Figure 6.11. In particular for the short GOP3 experiments the histogram shows a clear difference in the age distribution for the plain and the contiguous format.

Table 6.2 gives the experimentally determined neighborhood integrity (averaged over 200 trick pictures) when performing a fast playback of n=3.0±0.25%. Figure 6.12 plots the histogram of the neighborhood integrity distribution. The results are fully in accordance with the simulations of Chapter 4. The slice length only has a marginal effect on the neighborhood integrity. The contiguous format yields a much better integrity than the plain format. The improvement for the contiguous format is visually very significant: with the contiguous format the trick pictures are in the worst case built up in two distinct picture sections originating in different original pictures, with the plain format the trick picture can consist of as many as 10 different sections.

	GOP3 (pl)		*GOP3 (ct)*		*GOP13 (pl)*		*GOP13 (ct)*	
	average	*sdev*	*average*	*sdev*	*average*	*sdev*	*average*	*sdev*
3mb	0.84	0.06	0.93	0.04	0.84	0.06	0.94	0.03
15mb	0.82	0.06	0.94	0.03	0.83	0.06	0.95	0.02
45mb	0.81	0.06	0.94	0.03	0.82	0.07	0.95	0.02

Table 6.2: Average neighborhood integrity

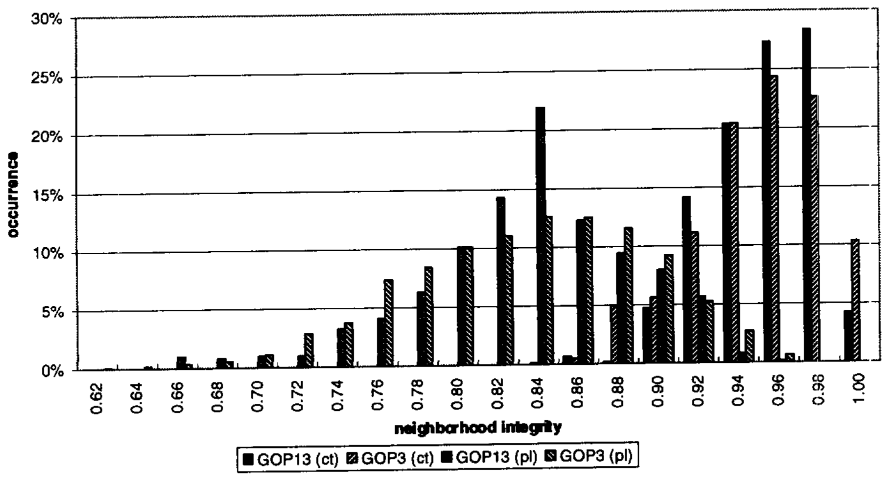


Figure 6.12: Neighborhood integrity for GOP3 and GOP13.

Figure 6.12 gives a histogram of the neighborhood integrity distribution for the 15mb version of the sequences; the results for the plain and contiguous format are plotted for both GOP3 and GOP13. Again it can be seen that the neighborhood integrity is not dependent upon the GOP structure. The contiguous formatting improves the integrity drastically for both GOP structures.

6.4 Dedicated Stream Fast Playback

Based on the contiguous map for the M4 machine with enhancement (E=2) and the methodology of Chapter 3 a dedicated fast playback map is designed, where the dedicated trick bit stream is guaranteed to be read during fast forward, (n=3.0) irrespective of the phase.

Using the minimum estimate for the Q-factor Q=0.54, the enhancement E=2 and the speedup factor n=3.0, we determine m=3, i.e. three copies of the dedicated fast playback stream are required in the contiguously re-mapped DFS. Experiments were performed with the verification hardware using the highest possible bit rate for the fast playback bit stream; the entire spare capacity was used for a single fast playback stream. As an example, consider the recording of a 9.6Mb/s normal play stream on a 12.5Mb/s recorder. With m=3, the bit rate of the fast playback signal is 2.9Mb/s.

Obviously, this bit rate is higher that what an actual system will provide for a single dedicated stream, as the available spare capacity will be divided over more speed-up factors.

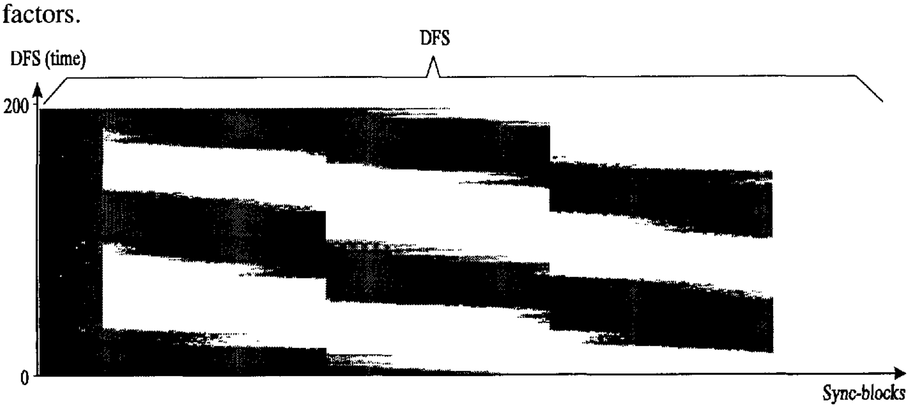


Figure 6.13: Re-mapped read mask with m=3

Figure 6.13 shows the resulting consecutive DFS read masks using the map designed for these parameters. The sync-blocks of the fast playback stream form the lower part of the DFS. The requirement is that this area is read for every consecutive DFS, such that the entire dedicated trick stream can be decoded during fast playback. It can be seen from the figure that in principle this objective is reached, except that at certain locations some sync-blocks are missing (white spots). The reason for this is that the designed map is too close to its theoretical limit which was established on the basis of the average read fraction, i.e. the number of copies of the fast playback stream is too close to the minimum. The assumption that the prediction based on the average Q-factor for the DFS holds for every individual track breaks down. For certain scanner to tape phases only the edge of the burst reads the fast playback area. A burst which is a little smaller will cause the effect of a missed sync-block.

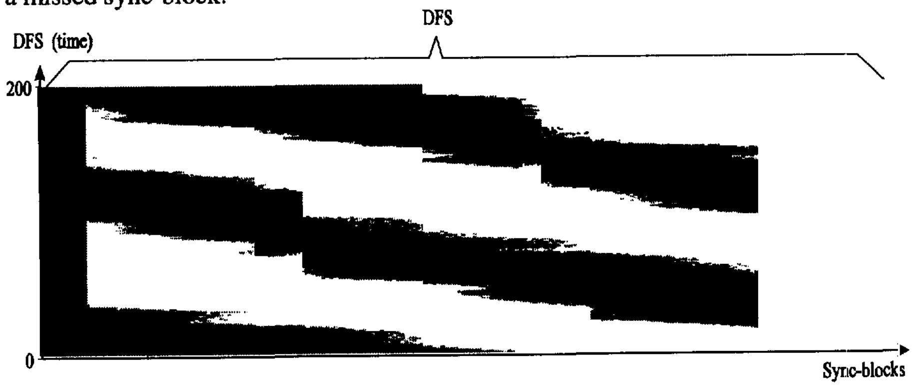


Figure 6.14: Re-mapped read mask with m=4

When a map is designed for m=4, such that 4 copies of a dedicated fast playback stream will exist on tape, then an extra margin exists. Figure 6.14 shows the corresponding map, where now no sync-blocks are missing from the 200 consecutive fast playback areas.

7. Discussion

In this thesis two different approaches to fast playback of helical scan recorded MPEG video have been studied and their performance has been evaluated. On one hand the rudimentary fast playback approach reconstructs a fast playback picture using the recorded normal play bit stream, on the other hand the dedicated fast playback approach uses a separately recorded fast playback bit stream. In this discussion we will take a broader perspective on the two studied approaches and discriminate between them.

We have introduced the rudimentary fast playback approach as a simple method where no special recording adaptation is necessary. This is an attractive feature which allows this approach to serve as a fall back method of performing fast playback when no special fast playback measures have been taken by a low complexity recording adaptation interface. Consequently, even when a recording was made by a low end recorder, still some fast playback is possible.
With the rudimentary fast playback approach, the implementation complexity of fast playback is placed at the playback side and the playback adaptation interface will require a bit stream validator. In the worst case, such a validator is as complex as the one discussed in Chapter 4. Depending on the capability of the decoder to decode partial picture structures, some of the functionality of the playback adaptation interface may be moved to the decoder. By using a bit stream validator to reconstruct a valid MPEG bit stream, the pause and slow motion trick modes are inherently handled correctly by the validator. It is therefore not necessary to handle these two trick modes differently from the fast playback trick modes.
Given the objective of the rudimentary fast playback to avoid specific fast playback measures during recording, the question arises how the contiguous format used in this thesis fits into this scheme. We have used the contiguous formatting of the normal play stream to enhance the fast playback behavior by changing the read mask; the recovered burst length is increased and the amount of bursts are decreased. In the hardware developed for validation purposes this was the only available method to change the nature of the fast playback read mask. The usage of the contiguous mapping has allowed us to analyse the effect of the specific read mask on the fast playback performance and it gives an indication of what performance increases are possible when the read mask is modified.
A practical recorder that that does not take the possibility of fast playback into account is not expected to perform such a contiguous mapping. In this case a comparable effect can however be reached by performing head actuation during fast playback. With head actuation, the head position on the scanner will be modulated to recover more of the bit stream of a single track pair. Performing head actuation thus requires a more complex playback bit machine to achieve simular advantages as those of the contiguous format, namely improved neighborhood integrity and a decreased macroblock picture age. Contrary to the contiguous format approach,

head actuation does allow the read mask to be improved for many different speedup factors at the same time.
An important drawback of the rudimentary fast playback approach is caused by the requirements and restrictions posed on the MPEG bit stream syntax. Some limitations are posed on the allowed variability of the syntax. More importantly there is the requirement that the intra slices are flagged separately in the slice header. Without this flagging rudimentary fast playback is not possible. In the concept of a low complexity recording adaptation interface, where no operations are performed on the bit stream prior to recording, this requires the encoder to perform the intra slice flagging itself. As the encoder will be located at the video source, e.g. the broadcaster, it is not clear at this point to what extent this small overhead (less than 0.2%) will be included. The situation may thus arise that for certain recorded sequences no fast playback is possible.
The resulting picture quality of the rudimentary fast playback video is enough for the purpose of performing a visual search but it is inferior to he fast playback quality of existing analog recorders. Especially for sequences with a high amount of motion and many scene cuts the coherance of the individual fast playback pictures often is low. This is a strong motivation consider dedicated stream fast playback.

The dedicated stream fast playback approach is much more flexible and generic as it does not pose any constraints on the allowed encoder syntax. In this approach most of the complexity of fast playback is placed at the recording side. The recording adaptation interface needs to transcode the received bit stream to obtain a dedicated fast playback stream. This dedicated stream is formatted in the spare capacity on the tape. The transcoding requires the adaptation interface to have some minimal knowledge of the MPEG syntax; the approach with the lowest complexity is the codeword extraction method described in Chapter 5.
The recording adaptation interface must format the dedicated fast playback bit stream such that it is guaranteed to be read during fast playback and it can utilize the freedom of the bit rate allocation and the formatting of the multiple copies to achieve the desired quality to robustness trade-off.
In order to assure interoperability among different recorders the multiplexing of the dedicated fast playback bit streams with the normal play bit stream on tape must be standardized. For the dedicated fast playback approach discussed in this thesis, where no scanner-to-tape phase locking exists during fast playback, there is no need to standardize the physical tape position of the fast playback stream. By specifying the multiplex and packet numbering any decoder will be able to extract the appropriate stream during playback.
It would be attractive to base the standardized multiplex definition for tape on the MPEG transport stream definition. In an MPEG digital video environment it can be expected that both the recorder adaptation interface and the decoder will need to have the functionality of a transport stream demultiplexer such that specific programs can be selected. Using the same transport stream multiplex for the

dedicated fast playback streams would then incorporate the dedicated fast playback streams in a seamless manner.
As far as the dedicated fast playback video bit stream itself is concerned, the only requirement is that it is a valid MPEG bit stream of the same profile as the normal play stream. The transcoding method does not need to form part of the standard and it may be a discriminating feature of MPEG recorders of different vendors; a high end recorder could add a fully transcoded dedicated trick stream to the recording, a low end recorder could add a simple zonal codeword extraction signal to the recording. Both recordings would be interchangeable for playback on either a high end or a low end recorder.

Both the rudimentary fast playback approach and the dedicated trick stream fast playback approach have been validated using a hardware demonstrator. The rudimentary fast playback performance was as predicted by the simulations.
A more important conclusion from the validation is that the model-based tape format for dedicated fast playback streams performs well with an actual recorder. The attractive feature of the format design method is that the robustness can be increased by using an extra copy of the dedicated stream on tape. Therefore a significant tolerance exists for the deviation of the actual system from the model and even for the error in the speedup factor due to the lack of a speed-lock. It is important to note that with respect to the decoder synchronization, this speed tolerance may be much smaller.
For an actual recording system our method to increase the robustness may not be the preferred approach. Instead, the second level of error correction (vertical track based error correction), which now is only effective during normal play, should also be used for fast playback. In a system with one or more fast playback bit streams the existing vertical error correction should therefore be modified to operate only on the normal play area of the track. A different second level error correction should be added to the separate fast playback streams. Conceptually the second level error correction could thus be moved to the outside of the mapping interface such that the same type of error correction can be applied to the individual streams.

When merging the principles of the rudimentary and the dedicated stream fast playback approaches, then new fast playback solutions arise. However, there does not seem to be any motivation for using such a hybrid approach. If nothing is done during recording and the entire problem of fast playback is solved by the playback adaptation interface (i.e. rudimentary trick modes) then only a limited fast playback quality will be possible. As soon as some functionality is added to the recording adaptation interface, then the transition to the full dedicated stream approach is quickly made. The generic nature of this dedicated stream approach, with many degrees of freedom in the complexity to quality trade-off, will make it preferable to hybrid solutions.

In this thesis the subject of fast playback of MPEG video was studied for the particular case of helical scan recording. It is interesting to consider if the discussed approaches are more generally applicable to different types of recorders.
It can be expected that many digital recorders, or more generally digital storage media that do not support random access, employ dedicated fast playback streams to support fast playback at different speeds. Often many different playback speeds need to be supported such that the dedicated stream approach would entail a significant overhead storage requirement if no transcoding were performed. The dedicated fast playback transcoding approach for the helical scan recorder and in particular the codeword extraction approach, can therefore be expected to be widely applicable to many storage systems.
The rudimentary fast playback evaluation is rather specific to the case of helical scan recording. In particular the read mask, on which our entire evaluation is based, is very recorder specific. If the principle of rudimentary fast playback is applicable to other storage devices then it is sure that a different read mask will be recovered. We can thus conclude that, where the principle of rudimentary fast playback may have a wider field of applicability, our specific results on this type of fast playback are limited to the helical scan recorder.

Finally we would like to conclude with the observation that MPEG is not as suitable to tape recording as is suggested by the standard. In particular the title of the MPEG-1 standard "Coding of moving pictures ... for digital storage media..." [MPEG1V] is misleading in this respect. With MPEG, trick modes can easily be supported by storage media that have random access possibilities. For a sequential storage device like the helical scan recorder this is however more complicated. The essential recording requirements have been included as optional tools only. This is perfectly in accordance with the toolkit approach of MPEG. However, this toolkit approach may be a drawback to the interopperability of different applications that individually would make different toolkit choices.

References

[Ahme74] N. Ahmed, T. Natarjan, K.R. Rao, *Discrete cosine transform*, IEEE Trans. on comp., Vol. 23, pp 90-93, Jan 1974.

[ATV93] Special Report, *Federal Comminucations Commission Advanced Television System Recommendation*, IEEE Trans. on Broadcasting, Vol 39, No 1, March 1993.

[Azad94] F. Azadegan et al., *Data-placement Procedure for Multi-speed Digital VCR*, IEEE Trans. on Consumer Electronics, Vol 40, No 3, August 1994, pp. 250-256.

[Berg71] T. Berger, *Rate Distortion Theory: A Mathematical basis for Data Compression,* Prentice-Hall, Englewood Cliffs, NJ, 1971.

[Bier88] M. Bierling, *Displacement estimation by hierarchical block matching*, in Proc. SPIE Conf. On Visual Comm. and Image Proc., Cambridge, Massachusetts, pp. 942-953, SPIE, Nov 9-11 1988.

[Boyc93] J M.Boyce, F. Lane, *Fast Scan Technology for Digital Video Tape Recorders*, IEEE Trans. on Consumer Electronics, Vol. 39, No. 3, Aug. 1993, pp. 186-191.

[Boyc94] J. M. Boyce, *Data Selection Strategies for Digital VCR Long Play Mode*, IEEE Trans. on Consumer Electronics, Vol 40, No 3, August 1994, pp. 269-273.

[CCIR601] ITU-R, Geneva, *CCIR Recommendation 601-1: Encoding Parameters of Digital Television for Studios*, 1986. Recommendations and Reports of the CCIR, Volume XI-Part 1, pp. 319-328.

[Chia95] L. Chiarglione, *The development of an integrated audiovisual coding standard: MPEG*, Proc. of the IEEE, Vol. 83, No. 2, pp 151-157, Feb 1995.

[DART12] RACE-2026 DART Deliverable #12, *Demonstration of a Verification Model for MPEG Formatted Services,* December 1994.

[Dries92] J. Driessen, *Motion Estimation for Digital Video.* PhD thesis, Delft University of Technology, 1992.

[DVB94] DVB, Television for the third millenium, DVB project office, ed. G.T. Waters, Aug. 1994.

[Ever63] H. Everett III, *Generalized Lagrange Multiplier Method for solving problems of optimum allocation of resources*, Operations research, vol11 pp 399-417, 1963.

[Feni95] C. Fenimore, et. al., *Summary Report on the Workshop on Advanced Digital Video in the National Information Infrastructure*, SMPTE Journal, pp 148-152, March 1995.

[Frim92] E.D. Frimout, J.N. Driessen, E.F. Deprettere, *Parallel Architecture for a Pel-Recursive Motion estimation Algorithm*, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 2, No. 2, June 1992, pp 159-168.

[Frim93] E.D. Frimout, J. Biemond, R. L. Lagendijk, *Forward rate control for MPEG recording*, Proc SPIE VCIP '93, Boston, Mass. Oct 1993.

[Frim94a] E.D. Frimout, J. Biemond, R. L. Lagendijk, *Trick mode solutions for MPEG tape recording*, Proc SPIE VCIP '94, Chicago Il. , Sept 1994.

[Frim94b] E.D. Frimout, J. Biemond, R. L. Lagendijk, *Tape recording of MPEG with Trick mode support,* Proc. Intl. Workshop on HDTV, Torino Italy, Okt '94.

[Frim95] E.D. Frimout, R.L. Lagendijk, J. Biemond, *Extraction of a dedicated fast playback MPEG bit stream*, Proc. SPIE VCIP'95, Taipei, Taiwan, May 1995.

[GA94] Grand Alliance, *Grand Alliance HDTV System Specification*, April 1994.

[GA95] Grand Alliance, *The U.S. HDTV standard*, IEEE Spectrum, pp 36-45, April 1995.

[Ghan93] M. Ghanbari, V. Seferidis, *Cell-loss Concealment in ATM Video Codecs*, IEEE Trans. on Circuits and Systems for Video Tech. Vol.3, No.3, June 1993, pp. 238-248.

[Giro87] B. Girod, *The efficiency of motion-compensating prediction for hybrid coding of video sequences*, IEEE J. Select. Areas Commun., vol. 5, pp. 1140-1154, Aug. 1987.

[Gonz91] C. Gonzales and E. Viscito, *Motion Video Adaptive Quantization in the transform domain*, IEEE Trans. Circuits Syst. Video Tech., Vol. 1, No. 4, December 1991, pp.374-378.

[H261] ITU-T, *Recommendation H.261: Video Codec for Audiovisual Services at p× 64 kbit/s*, March 1993.

[Hask72] B. Haskell, F. Mounts, J. Candy, *Interframe coding of videotelephone pictures*, Proceedings of the IEEE, Vol.60, No. 7, pp 792-800, July 1972.

[Hopk94] R. Hopkins, *Digital Terrestrial HDTV for North America: The Grand Alliance HDTV System*, IEEE Trans. on Consumer Electronics, Vol. 40, No 3, August 1994, pp 185-198.

[Jung94] K.H. Jung, C.W. Lee, J.H. Chang, *Error concealment technique using projection data for block-based image coding*, Proc SPIE VCIP '94, Chicago, Il. Sept. 1994

[Kauf91] P. Kauff, S. Rauthenberg, M. Hahn, *A DCT Coding Scheme for Digital HDTV Recording*, Intl Worksh. on HDTV '91 Sept 1991, Turin, Italy.

[Kees94] G. Keesman, "Bit-rate control for MPEG encoders", Signal Processing: *Image Communications*; Vol. 6, Nr. 6, pp 545-560, Feb. 1995.

[Kond91] T. Kondo, Y. Fujimori, H. Nakaya, *Adaptive Dynamic Range Coding Scheme for Future HDTV Digital VTR*, Intl Worksh. on HDTV '91 Sept 1991, Turin, Italy.

[Lane93] F. Lane, J. Boyce, Fast Forward/Fast Reverse for a Digital HDTV VCR, Intl. Worksh. on HDTV '93, Nov 1993, Ottawa Canada

[LeGa91] D. Le Gall, *MPEG: A video compression standard for multimedia applications*, Communications of the ACM; Vol. 34, pp 46-58, April 1991.

[LeGa92] D. Le Gall, *The MPEG video compression algorithm*, Sig. Proc.: Image Communication, vol. 4, pp. 129-140, Apr. 1992.

[Liou91] M. Liou, *Overview of the p*64 kbit/s video coding standard*, Communications of the ACM; Vol. 34, pp 60-63, April 1991.

[Look95] T. Lookabaugh, *Matching MPEG-2 to Applications*, SMPTE Journal, pp. 4-8, Jan. 1995.

[McLa93] D. L. McLaren, A. Gersho, *An HDTV Compression Scheme for Production-Quality Recording and High-Performance Shuttle Operation*, IEEE Trans. on Consumer Electronics, Vol 39, No 3, August 1993, pp175-185.

[MPEG1S] ISO/IEC 11172-1, IS, Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s - Part 1: System.

[MPEG1V] ISO/IEC 11172-2, International Standard, Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s - Part 2: Video.

[MPEG2A] IEC/ISO 13818-3 DIS General Coding of Moving Pictures and Associated Audio - Part 3: Audio.

[MPEG2C] IEC/ISO 13818-4 DIS General Coding of Moving Pictures and Associated Audio - Part 4: Conformance.

[MPEG2S] IEC/ISO 13818-1 DIS General Coding of Moving Pictures and Associated Audio - Part 1: System.

[MPEG2V] IEC/ISO 13818-2 DIS General Coding of Moving Pictures and Associated Audio - Part 2: Video.

[MPEGTM4] ISO-IEC JTC1/SC29/WG11, *Test Model 4 for MPEG2*, MPEG93/225b, Feb 1993.

[Muss85] H. Mussmann, P. Pirsch and H. Grallert, *Advances in Picture coding*, Proc. IEEE, vol. 73, pp.523-547, Apr. 1985.

[Nata77] T.R. Natarjan, N. Ahmed, *On interframe transform coding*, IEEE Trans. on communications, Vol.25, pp 1323-1329, Nov 1977.

[Okam93] H. Okamoto et. al., *A Consumer Digital VCR for Advanced Television*, IEEE Transactions on Consumer Electronics, Vol. 39, No. 3, August 1993, pp.199-204.

[Okub92] S. Okubo, *Requirements for high quality video coding standards*, Sig. Proc.: Image Communication, vol. 4, pp. 141-151, Apr. 1992.

[Okub93] S. Okubo, K. McCann, A. Lippman, *MPEG-2 Requirements, Profiles, Performance Verification*, Intl Worksh. on HDTV '93, Nov 1993, Ottawa Canada.

[Pire93] C. Pires, A. Rinaudo, S. Battista, S. Dal Lago, *The Influence of Channel Errors on a Video Decoder Syntax and Hardware Architecture for Broadcasting of TV/HDTV Applications*, Intl Worksh. on HDTV '93, Nov 1993, Ottawa Canada.

[Puri91] A. Puri and R. Aravind, *Motion-Compensated Video Coding with Adaptive Perceptual Quantization,* IEEE Trans. Circuits Syst. Video Tech., Vol. 1, No. 4, December 1991, pp. 351-361.

[Ramc93] K. Ramchandran, A. Ortega and M. Vetterli, *Bit Allocation for Dependent Quantization with Applications to MPEG Video Coders*, Internat. Conf. Acoust. Speech and Signal Process. '93, Minneapolis, U.S.A., April 1993, p 381-384.

[Ramc94] K. Ramchandran, M. Vetterli, *Rate-Distortion Optimal Fast Thresholding with complete JPEG/MPEG Decoder Compatibility*, IEEE Trans. Image Processing, Vol. 3, No. 5, September 1994.

[Reim94] U. Reimers, *Concept of a European System for the Transmission of Coded Television Signals via Satellite*, SMPTE Journal, pp 741-747, Nov. 1994.

[Seza93] M.I. Sezan, R.L. Lagendijk, *Motion Analysis and Image Sequence Processing*, ed., Kluwer Academic publishers, 1993.

[Shli94] S. Shlien, *Guide to MPEG-1 Audio Standard*, IEEE Trans on Broadcasting, Vol 40, No 4, December 1994, pp 206-218.

[Shoh88] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers", IEEE Trans. Acoust. Speech, Signal Proc. vol. 36, pp. 1445-1453, Sept. 1988.

[ST3500] SGS-Thomson STi3500A MPEG-2/CCIR 601 Video Decoder, *System Description*, 1994.

[Stoj95] M.M. Stojancic, C. Ngai, *Architecture and VLSI Implementation of the MPEG-2: MP@ML Video Decoding Process,* SMPTE Journal, pp. 62-72, Feb. 1995.

[Suga88] H. Sugaya and K. Yokoyama, Chapter 2 from C.D. Mee and E.D. Daniel, *Magnetic Recording*, Vol. 3, McGraw-Hill, New York, 1988.

[Sun92] H. Sun, K. Challapali, J. Zdepski, *Error Concealment in Digital Simulcast AD-HDTV Decoder,* IEEE Trans.on Consumer Electronics, Vol 38, No 3, August 1992, pp 108-118.

[With92a] P.H.N. de With, *Data Compression Systems for Home-Use Digital Video Recording*, IEEE Journal on Sel. Areas in Comm. Vol 10. No 1, Jan. 92, pp 97-121.

[With92b] P.H.N. de With, *Data Compression Techniques for Digital Video Recording,* PhD thesis, Delft University of Technology, 1992.

[With93a] P.H.N. de With, A.M.A. Rijckaert, H. -W. Keesen, J. Kaaden, C. Opelt, *An Experimental Digital Consumer HDTV Recorder using MC-DCT Video Compression,* IEEE Trans. on Consumer Electronics, Vol 39, No 4, Nov 1993, pp711-722.

[With93b] P.H. de With and S.J. Nijssen, *A buffer regulation Concept for MC-DCT Systems Tuning to Constant Quantization,* Benelux Symp. Inf. Theory, 17-18 May 93, Veldhoven, The Netherlands.

[Wong94] C. Wong, A.J.M. Denissen, *Channel Investigations of a Helical-scan Digital Video Recorder at Normal and Trick-mode Playback,* Intr. Conf. on Storage and Recording Systems, April 94, Keele Univ., UK.

[Wu91] S.-W. Wu and A. Gersho, *Rate-Constrained Optimal Block-Adaptive Coding for Digital Tape Recording of HDTV,* IEEE Trans. Circuit and Syst. Video Tech., Vol. 1, No. 1, March 1991, pp.100-112.

[Yana93a] N. Yanagihara, C. Siu, K. Kanota, Y. Kubota, *A Video Coding Scheme with a High Compression Ration for Consumer Digital VCR's,* IEEE Trans. on Consumer Electronics, Vol 39, No 3, August 1993, pp 192-198.

[Yana93b] N. Yanagihara, C. Chan, Y. Kubota, *A Recording Method of ATV data on a Consumer Digital VCR,* Intl Worksh. on HDTV '93, Nov 1993, Ottawa Canada.

[Zhu93] Q.F. Zhu, Y. Wang, L. Shaw, *Coding and cell-loss recovery in dct-based packet video,* IEEE Trans. on Circuits and Systems for Video Tech. Vol.3, No.3, June 1993, pp. 248-258.

Samenvatting

De laatste jaren is er een enorme toename is de beschikbaarheid van hardware voor digitale video compressie. In het bijzonder zijn sinds kort MPEG decoders beschikbaar waarbij de hele decodeer stap door één enkele chip wordt uitgevoerd. Hierdoor wordt de ontwikkeling van een breed scala aan digitale videodiensten gestimuleerd. Enerzijds worden door de integratie van digitale video met communicatie netwerken nieuwe interactieve video toepassingen geïmplementeerd. Voorbeelden hiervan zijn video op aanvraag (VOD: video on demand) en het over een netwerk zoeken in video-databases. Anderzijds worden van de bestaande videodiensten, waar TV de belangrijkste van is, digitale varianten geïmplementeerd. Het belangrijkste voordeel van de digitale video is hier de lagere vereiste bandbreedte.
Men kan dan ook verwachten dat in de nabije toekomst digitale TV transmissie een steeds belangrijkere rol zal gaan spelen en geleidelijk aan de analoge transmissie zal vervangen. Door de mogelijkheid om video met een goede beeldkwaliteit real-time te decoderen zal MPEG-2 op korte termijn de meest gebruikte digitale compressie standaard zijn. Dit blijkt vooral uit de intussen vastgelegde standaarden voor digitale video diensten.
Binnen het digitale video scenario van de toekomst neemt een digitale recorder een centrale plaats in. Voor opnemen van een of meer digitaal gecomprimeerde video signalen is *helical scan recording* een toepasselijke technologie. Het doel van het DART (Digital dAta Recording Terminal) project, dat het raamwerk vormt voor dit proefschrift, is om een helical scan recorder te ontwikkelen voor het brede scala aan te verwachten digitale video diensten.
Vanuit het standpunt van de consument moet een nieuwe digitale video recorder naast het normaal afspelen ook de welbekende trick modes hebben als pauze, slow motion, en versneld afspelen (*fast playback*). Een video beeld weergeven tijdens fast playback is bijzonder ingewikkeld voor gecomprimeerde digitale video. Dit komt doordat bij een helical scan recorder na het versnellen van de band slechts kleine secties van de opgenomen bitstroom teruggevonden worden. De fysieke positie van deze bitstroom secties op de band is in het algemeen slecht voorspelbaar en hangt af van de fase tussen de scanner en de band. Indien de bitstroom secties willekeurige happen uit de bitstroom vormen kunnen ze in het algemeen niet individueel gedecodeerd worden. Verder is niet voorspelbaar welke delen van het videobeeld ze representeren.
Het is dus noodzakelijk om een methode te ontwikkelen om versneld afspelen van een MPEG gecomprimeerde video opname mogelijk te maken. Een belangrijke eis binnen het DART project is dat de ontwikkelde methode geverifieerd kan worden met een experimentele digitale recorder.

In dit proefschrift beschouwen wij MPEG gecomprimeerde standaard resolutie video van 10Mb/s. Hiervoor wordt een 12.5Mb/s recorder gebruikt. Twee

verschillende benaderingen voor fast playback van een digitale video opname worden beschouwd, namelijk *rudimentary fast playback* en *dedicated-stream fast playback.*

In de eerste benadering voor het versneld afspelen, die we de rudimentaire benadering noemen, wordt de bitstroom rechtstreeks opgenomen. De hele opgenomen stroom kan voor het normaal afspelen (*normal play*) gebruikt worden. Voor fast playback gebruiken we de toevallig teruggevonden bitstroom secties uit de normal play stroom om een versneld beeld te reconstrueren. Het grootste deel van de complexiteit van fast playback bevindt zich in deze benadering aan de afspeelkant. De adaptatie laag, tussen de recorder en de decoder, moet tijdens het afspelen de individueel decodeerbare delen (*intra slices*) verzamelen en hieruit een valide MPEG bitstroom reconstrueren.
De visuele kwaliteit van dit type fast playback is afhankelijk van het aantal intra slices in het normal play signaal en de grootte van de teruggewonnen secties. Het aantal intra slices bepaalt de gemiddelde tijdsduur tussen de bijwerkingen van de beelden van de gereconstrueerde fast playback videosequentie. Het is dus wenselijk om een bitstroom te hebben met veel intra slices en om zo groot mogelijke secties in een keer terug te winnen. In het toekomst scenario van digitale TV zal de recorder echter geen invloed kunnen uitoefenen op de exacte methode van encoderen die door de encoder gehanteerd wordt. Om toch een verbetering van de beeldkwaliteit te bewerkstelligen kan bij de opname een toepasselijke afspiegeling van de bitstroom op de band uitgevoerd worden zodat bij fast playback langere secties teruggewonnen worden. Deze langere secties representeren grotere delen van het beeld. Het bijwerken van grotere beelddelen in een keer resulteert in een significant betere beeldkwaliteit.

De tweede fast playback methode maakt gebruik van een specifiek hiervoor opgenomen stroom, de dedicated-stream. De vrije ruimte op de band kan gebruikt worden om deze dedicated-stream op te nemen naast de bestaande normal play stroom. De dedicated-stream kan zonder meer door een standaard MPEG decoder gedecodeerd worden zodat bij het afspelen slechts een zeer beperkte adaptatie nodig is. In plaats hiervan wordt het probleem van het versneld afspelen nu naar de opname adaptatielaag verschoven.
Om te garanderen dat de dedicated-stream bij het versneld afspelen teruggewonnen kan worden voor elke mogelijke fase tussen de scanner en de band is het noodzakelijk om deze een aantal keer op te nemen. Bijvoorbeeld vereist een typisch systeem, bij de snelheid n=3.0 ten opzichte van de normal play snelheid, dat drie kopijen van dezelfde dedicated-stream opgenomen worden. De manier waarop de kopijen van de dedicated-stream met de normal play stroom ge-multiplexed worden op de band bepaalt de *tape-format*. We hebben een methodologie ontwikkeld om, op basis van een model van de teruggewonnen data secties voor gegeven snelheid en fase, een geschikt tape format te ontwerpen voor specifieke versnellingsfactoren. De

methodologie kan uitgebreid worden om meerdere stromen voor verschillende snelheidsfactoren op te nemen.
De dedicated-stream voor het versneld afspelen moet een MPEG stroom zijn die slechts een kleine bandbreedte vereist (~1 Mb/s). Er bestaan veel mogelijkheden om deze dedicated-stream uit de normal play stroom te transcoderen. In dit proefschrift stellen wij een aantal codewoord extractie methoden voor. In het bijzonder werd een nieuwe extractiemethode ontwikkeld die de optimale codewoord extractiemethode benadert terwijl de complexiteit slechts weinig hoger is dan die van de traditionele vaste zone extractie methode die in de literatuur beschreven wordt.

Beide methoden om versneld af te spelen zijn geverifieerd met behulp van de hardware verficatie recorder die in het DART project is ontwikkeld en de speciaal voor dit doel ontworpen hardware adaptatie schakelingen. Het encoderen en decoderen van de MPEG video werd voor deze verificatie off-line uitgevoerd. Voor beide methoden bevestigt de validatie de resultaten van de simulaties. In het bijzonder blijkt na de validatie dat het tape-format voor de specifieke versnelde stroom, dat ontworpen werd op basis van een model, werkt zoals verwacht. De methodologie voor het ontwerpen van een tape-format is flexibel genoeg om het verloop van de snelheidsparameters te compenseren door een additionele kopij van de specifieke stroom op te nemen. Deze extra robuustheid laat vanzelfsprekend een kleinere bandbreedte over voor het specifieke signaal.

Acknowledgments

It is a pleasure to be able to thank those who have contributed to the work reported in this thesis.
First of all I would like to acknowledge the DART project for providing the framework for this thesis and the means for performing the experimental evaluation of the solutions. I would like to express my appreciation for the interesting collaboration to all project participants. In particular I wish to thank the project management: Dr. E. Persoon, Dipl.-Ing. W. Bachnick, Dr. Greiner, Prof. J. Biemond and Ir. E. Ideler. Furthermore I would like to acknowledge the European RACE-II program for the support of the DART project.
The collaboration towards the demonstrators has been most rewarding and it has had a significant effect on the contents of this thesis. I would therefore like to thank the Information Theory Group hardware team: Silvian Kindt, Pepijn Sitter, Peter Lems, Hans Verschuur and Ad de Ridder for the interesting brainstorming sessions which were held at various stages during the development. I have particular appreciation for Silvian who has been able to cope with various last minute changes in the requirement without ever loosing his patience. I also wish to thank Albert Rijckaert and Calto Wong from the Philips research magnetic recording group for their patience in explaining magnetic tape recording and the basics of interfacing to the DVT hardware.
I thank my promotor, Jan Biemond, for his support of my work and for allowing me the opportunity to be involved with all aspects of the project. My supervisor Inald Lagendijk has been very supportive in reviewing my work and his thorough feedback has allowed me to improve significantly on the original draft of this thesis.
I would like to thank all the members of the Information Theory Group for the pleasant working environment. I particularly enjoyed the 11:45 lunches with office 11.23, Ruggero Franich, Patrick van der Meer and Stefan Westen. I would also like to thank Annett Bosch for being the ever ready sunshine of the group.
Finally, I am highly indebted to Inge for her support throughout the past four years.

Curriculum Vitae

Emmanuel David Frimout was born in Ukkel, Belgium, on April 13th, 1968. In 1986 he graduated from the International Baccalaureate program of the Rijnlands Lyceum Oegstgeest. He acquired his degree in Electrical Engineering, Cum Laude, from the Delft University of Technology in 1991. During the course of his study he performed a year of practical experience work at Teledyne Brown Engineering (Alabama, US), the National Aeronautics and Space Administration (Alabama, US) and the Office Nationale des Etudes et Recherches Aerospatiales (Paris, France) respectively. His master's project was jointly carried out at the Information Theory and the Network Theory group and his thesis was entitled: "Parallel Architecture for a Pel-recursive Motion Estimation Algorithm".

From July 1991 until June 1995 he worked at the Information Theory Group on the RACE-DART project, sponsored by the European Union. For this project he has researched several aspects of MPEG recording and he has been responsible for the definition of one of the project's demonstrators.

Since July 1995 he has been working for the Advanced Development Center of Philips Sound and Vision where he is able to continue working in his field of interest.